Как я настраивал почтовый сервер [TEGU Enterprise](https://mbk-lab.ru/development/TEGU-enterprise/)

Эта инструкция написана для пользователей Windows не имеющих опыта работы в Linux.

Писалась эта инструкция и для себя, так как редко работаю с Linux и через некоторое время порядок действий забывается, для восстановления приходиться «лазить» по разным местам с информацией. Поэтому решил собрать всю информацию в одном месте.

Наверняка я что-то сделал не совсем правильно, поэтому буду признателен за поправки и советы от опытных пользователей Linux.

Сначала хотел развернуть сервер на «нашем» линуксе RED OS, но протестировав и сравнив с AlmaLinux, решил остановиться на последнем. RED OS не совсем корректно работала с сетевыми картами встроенными в мою материнскую плату. Для эксперимента был собран сервер на базе материнской платы от Supermicro **X13DEI**, двух двенадцати ядерных процессорах Intel(R) Xeon(R) Silver 4410Yи памятью 128 ГБ. Также в конфигурацию сервера входит аппаратный raid контроллер Adaptec SmartRAID 3100 серии. В BIOS материнской платы должна быть включена возможность виртуализации (на серверных системах включена по умолчанию).

Сама система будут устанавливаться на массиве raid1 из двух SSD дисков, а СУБД почты будет располагаться на массиве raid10 из дисков HDD. Объем этого массива должен быть не меньше «размер ящика пользователя» х «количество пользователей». Например, 1000 пользователей и объем ящика пользователя 20 ГБ, получаем 20 ГБ х 1000 = 20 ТБ.

Будут установлены сама система AlmaLinux 9 с поддержкой виртуализации и, для почтового сервера Tegu, три ВМ (виртуальные машины). Сам почтовый сервер Tegu, СУБД PostgreSQL и почтовый веб клиент RoundCubemail.

Вся информация по настройке почтового сервера Tegu взята с сайта поддержки Tegu

<https://project.mbk-lab.ru/>

Для настройки самой системы Almalinux использована информация с сайтов:

<https://www.server-world.info/en/note?os=AlmaLinux_9&p=download>

<https://wiki.crowncloud.net/index.php#AlmaLinux+9>

<https://redos.red-soft.ru/base/redos-8_0/8_0-install/8_0-sys-req/>

и других.

В первой части инструкции мы установим сам сервер Almalinux 9.

Во второй узнаем что такое удаленный доступ к серверу через веб интерфейс Cockpit, подключимся к серверу линух по SSH с помощью программы Putty ([PuTTY: Telnet/SSH Клиент](https://putty.org.ru/)) и установим Samba сервер и клиент на Almalinux 9 для простого обмена файлами между компьютером с Windows и нашим сервером. Так же дам ссылку на удобную программу с привычным для пользователей windows графическим интерфейсом для работы с файлами линух.

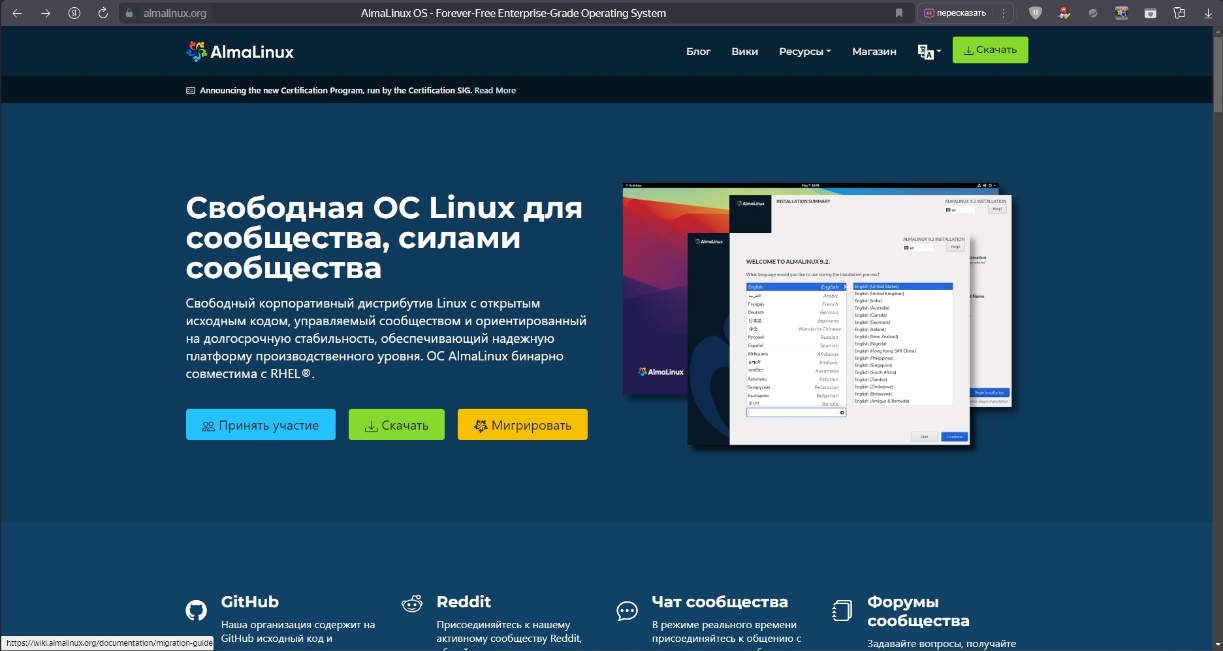
В третьей части настроим стандартный, дпя Windows, удаленный рабочий стол (RDP) для подключению к серверу Almalinux 9.

В четвертой части обновим ядро нашего сервера с v5 на v6.

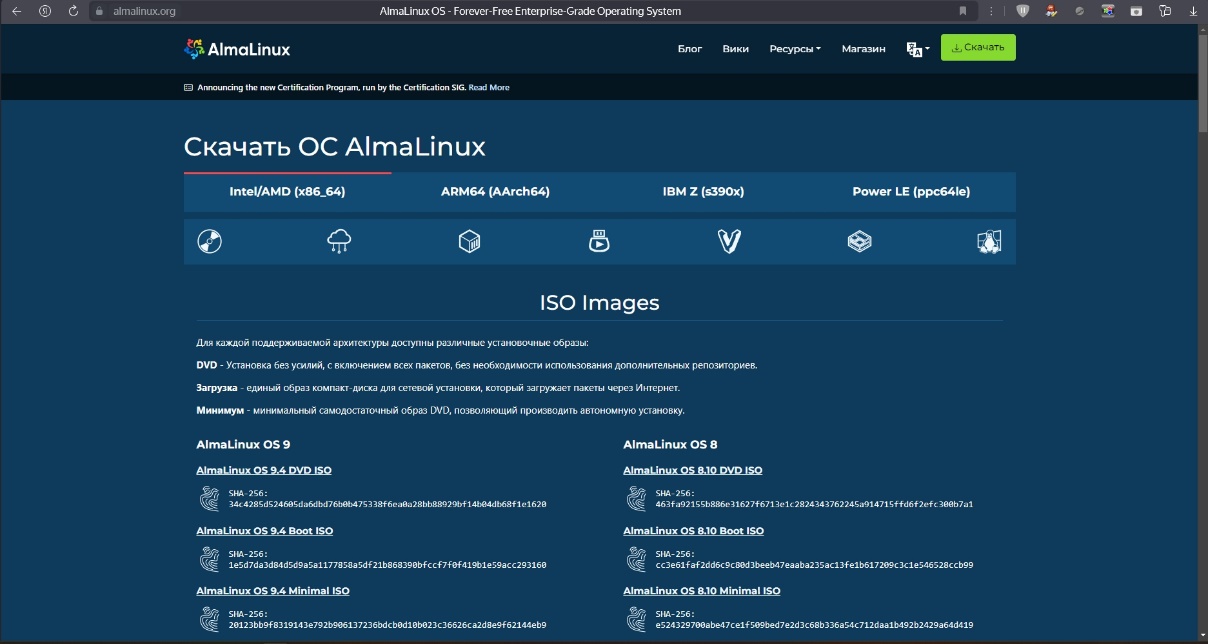
В пятой части перейдем к установке и начальной настройке KVM (виртуальных машин) необходимых для работы почтового сервера Tegu.

Часть первая. Установка сервера Almalinux.

Заходим на сайт <https://almalinux.org/ru/>



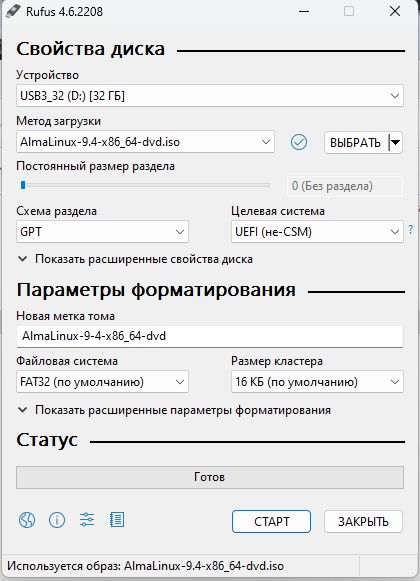
И нажимаем кнопку «Скачать»



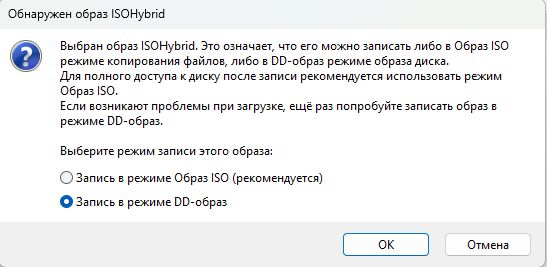
Выбираем [**AlmaLinux OS 9.4 DVD ISO**](https://repo.almalinux.org/almalinux/9.4/isos/x86_64/AlmaLinux-9.4-x86_64-dvd.iso) образ и сохраняем на компьютере.

Для подготовки загрузочной флэшки берем флэшку не менее 16 ГБ. Записывать образ будем программой Rufus <https://rufus.ie/ru/> Вставляем флэшку в компьютер и запускаем программу.

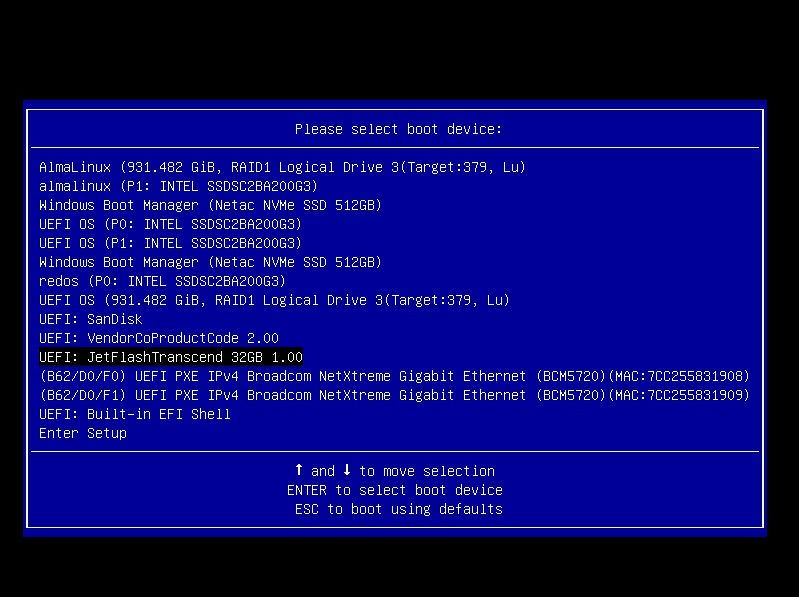
Так-как у меня сервер с БИОСом UEFI, то в программе делаем следующие настройки

****

После нажатия кнопки «СТАРТ» появляется следующее окно, выбираем «Запись в режиме DD-образ» и записываем флэшку. На этом подготовка загрузочного носителя завершена.



Вставляем флэшку в наш сервер и включаем его. В процессе запуска сервера выбираем режим входа в меню загрузки и выбираем нашу флэшку.

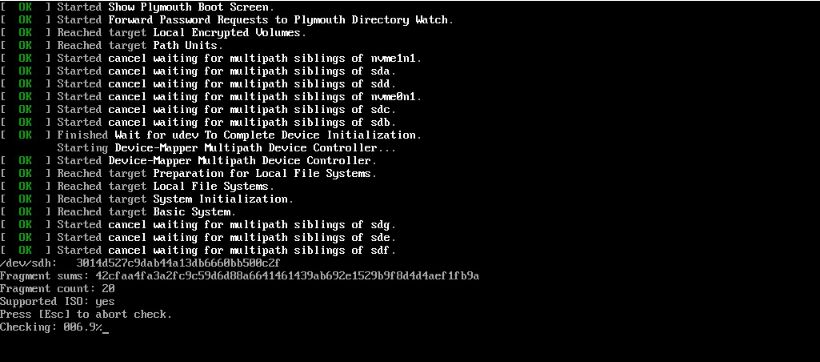


Происходит загрузка и появляется меню установки системы.

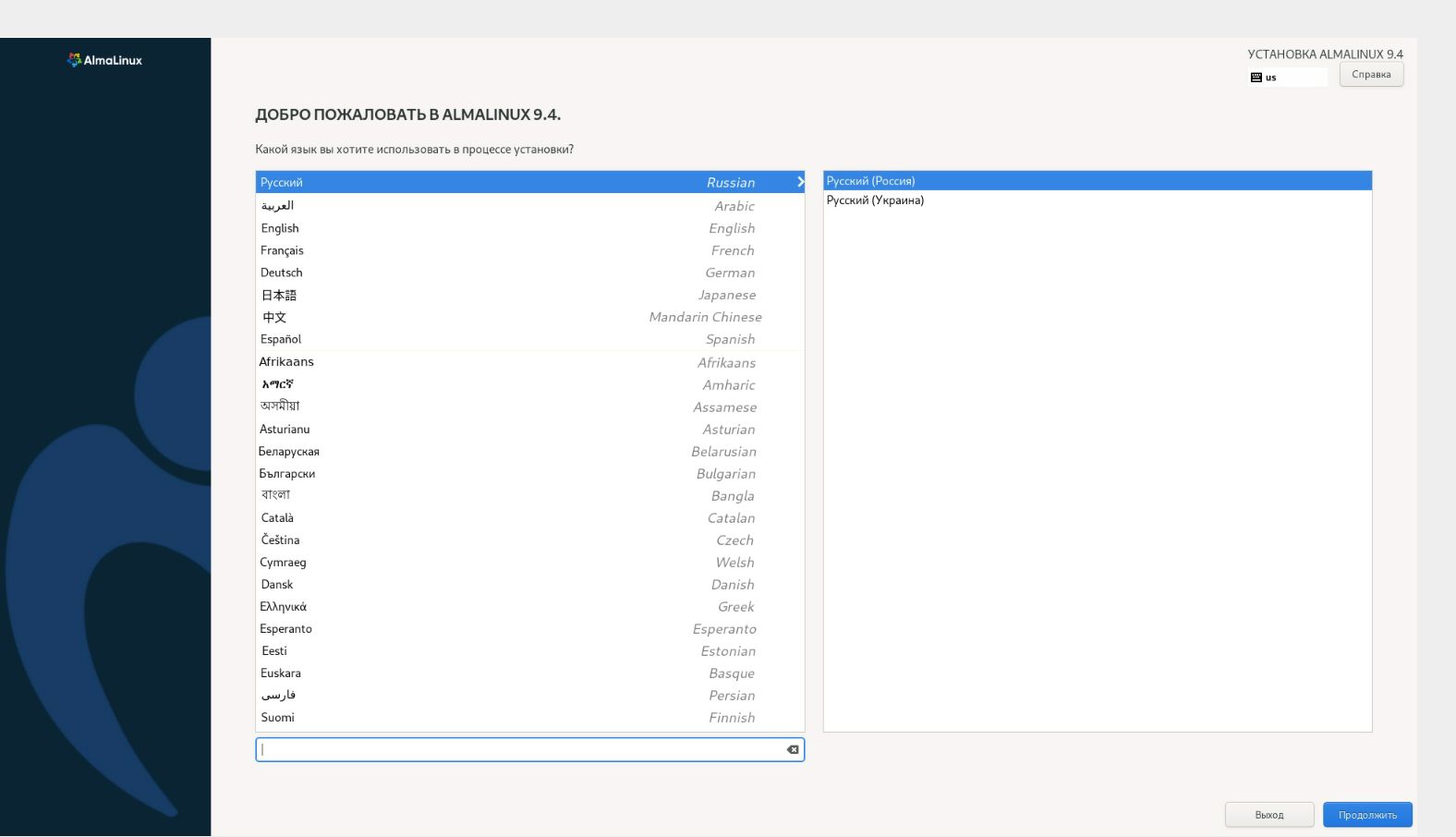
Выбираем проверить наш дистрибутив и начать установку



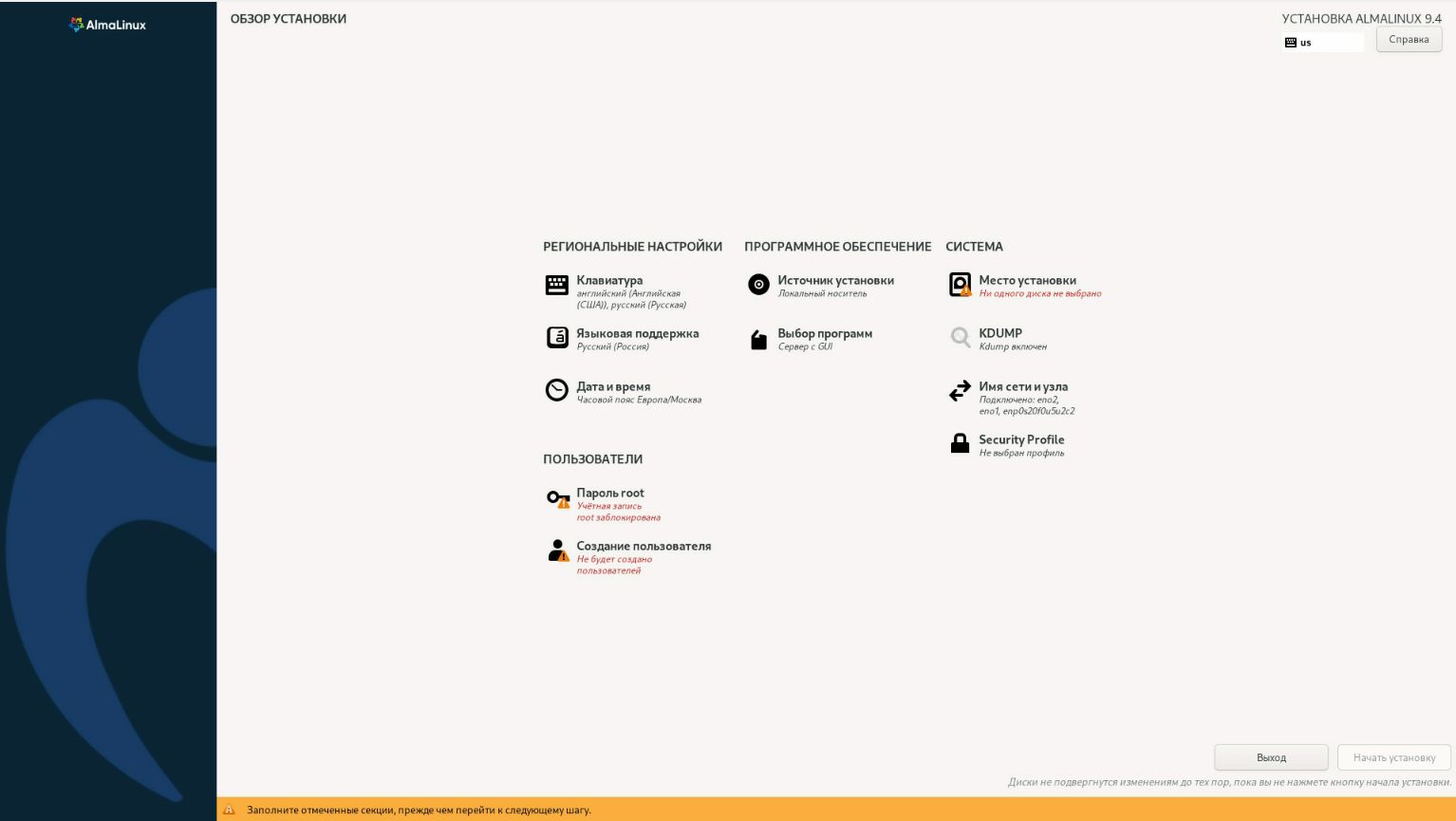
Начинается процесс установки



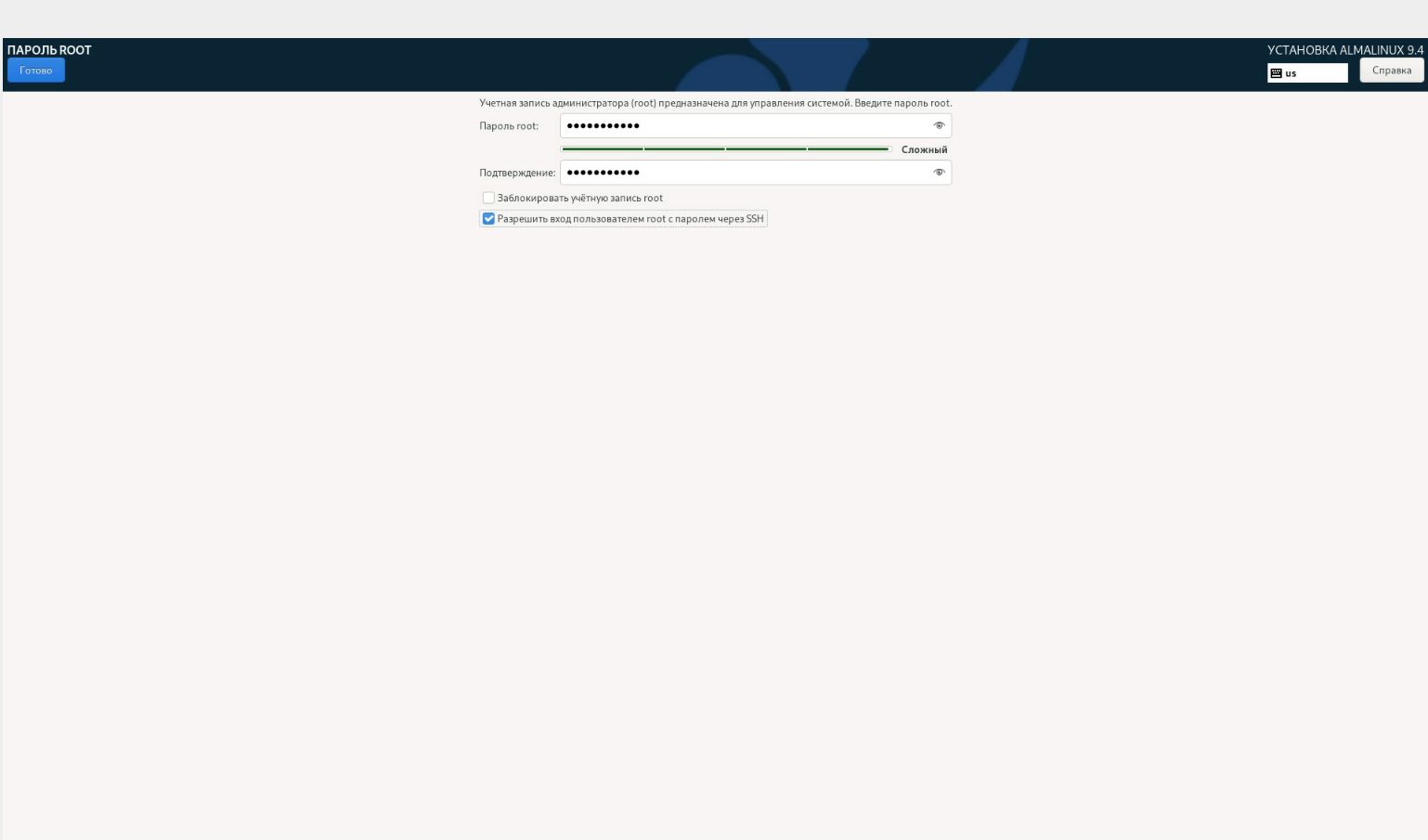
И открывается следующее меню



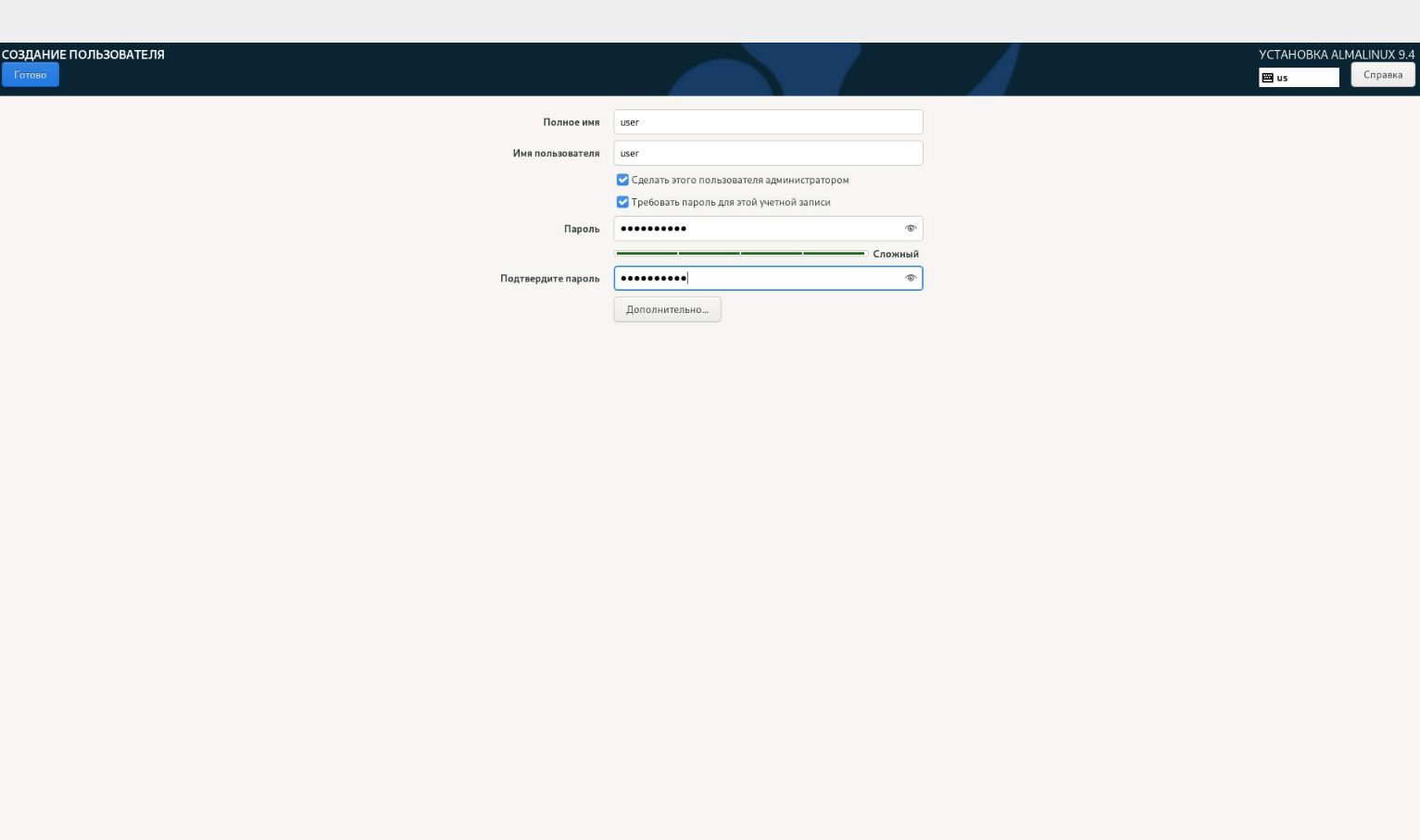
Нажимаем «Продолжить». Открывается следующее меню.



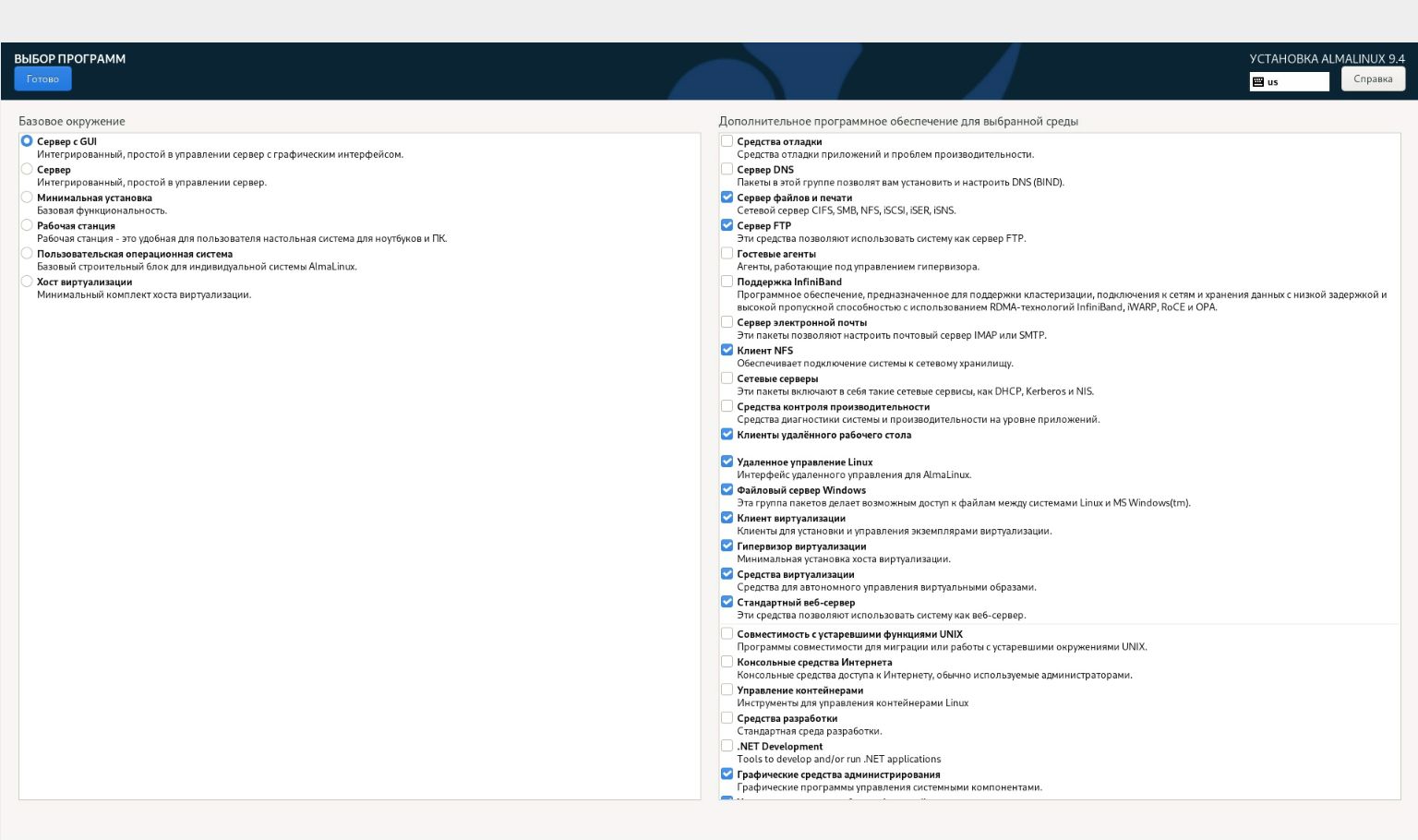
Выбираем «Пользователи» и создаем пользователя root (суперпользователь в системе линукс). Право на вход в сеанс SSH на рабочем сервере лучше не включать.

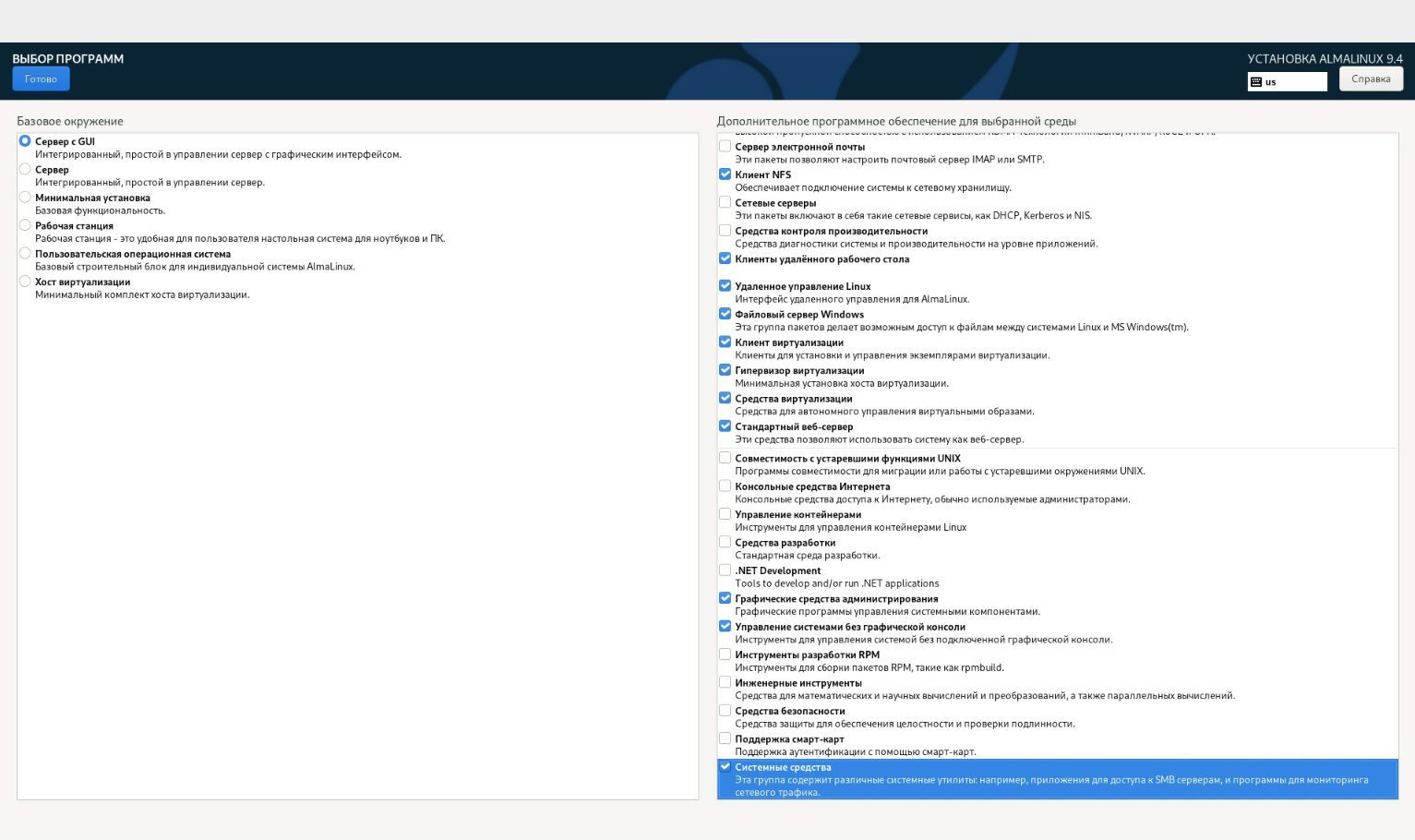


Далее добавляем пользователя с административными правами под которым мы будем работать на сервере. В примере ниже замените «user» на своего пользователя. Важно! В линуксе учетные записи пользователей, имена файлов чувствительны к регистру. Пользователь user и User это разные пользователи. Пользователей задавайте на «US» раскладке.

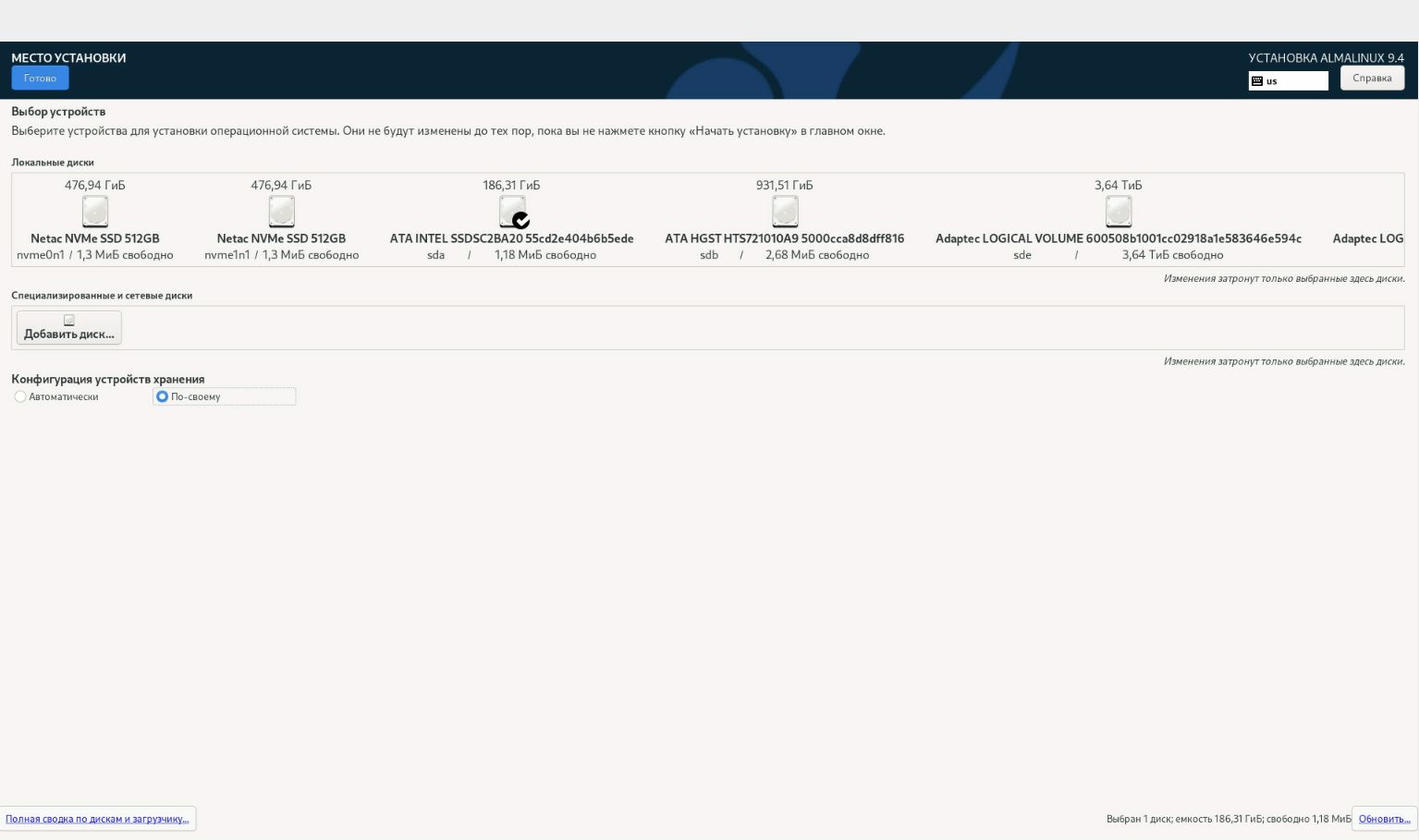


Далее переходим к выбору программ и выбираем как на рисунках ниже. Можно ограничить выбор и установить необходимые программы позже.

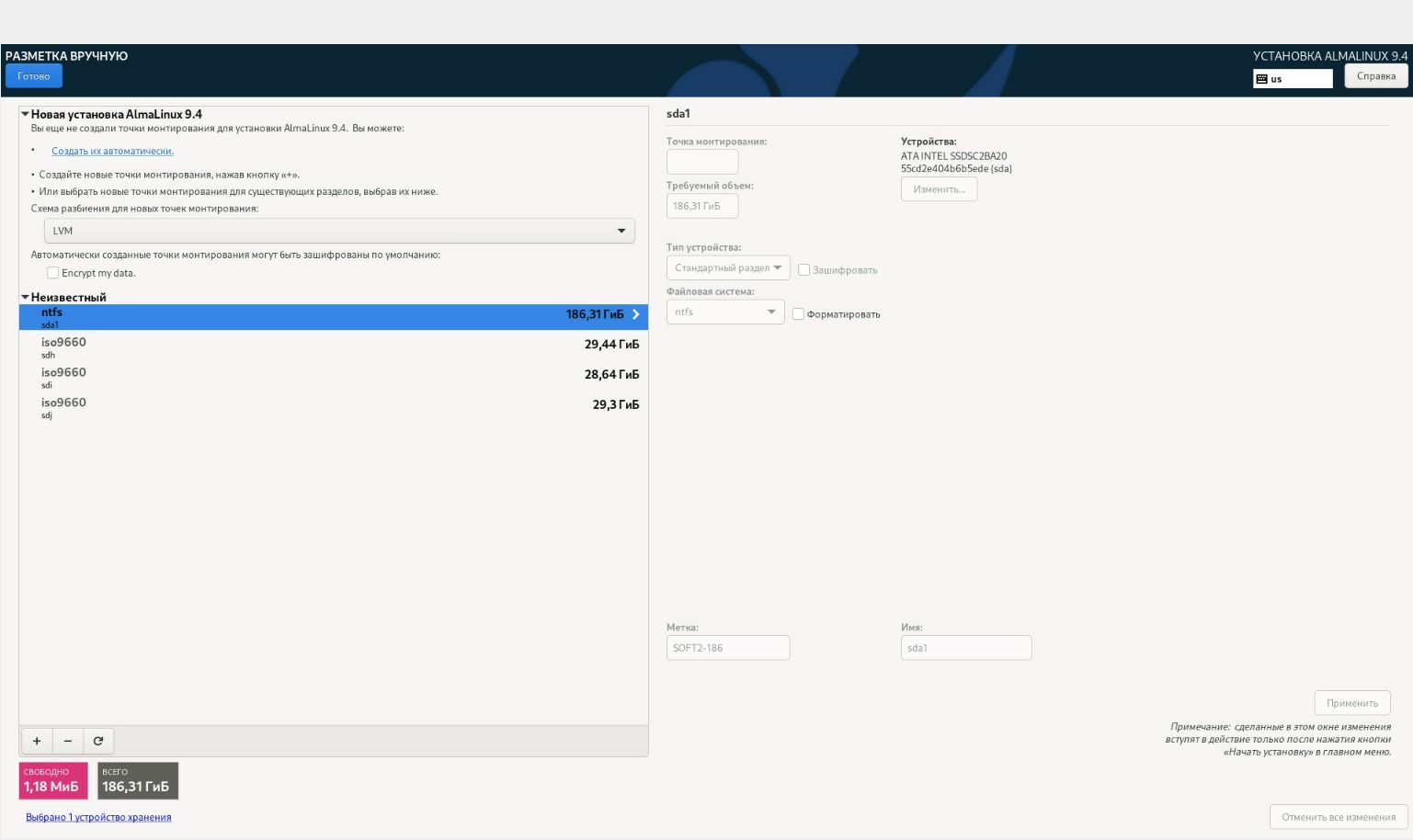




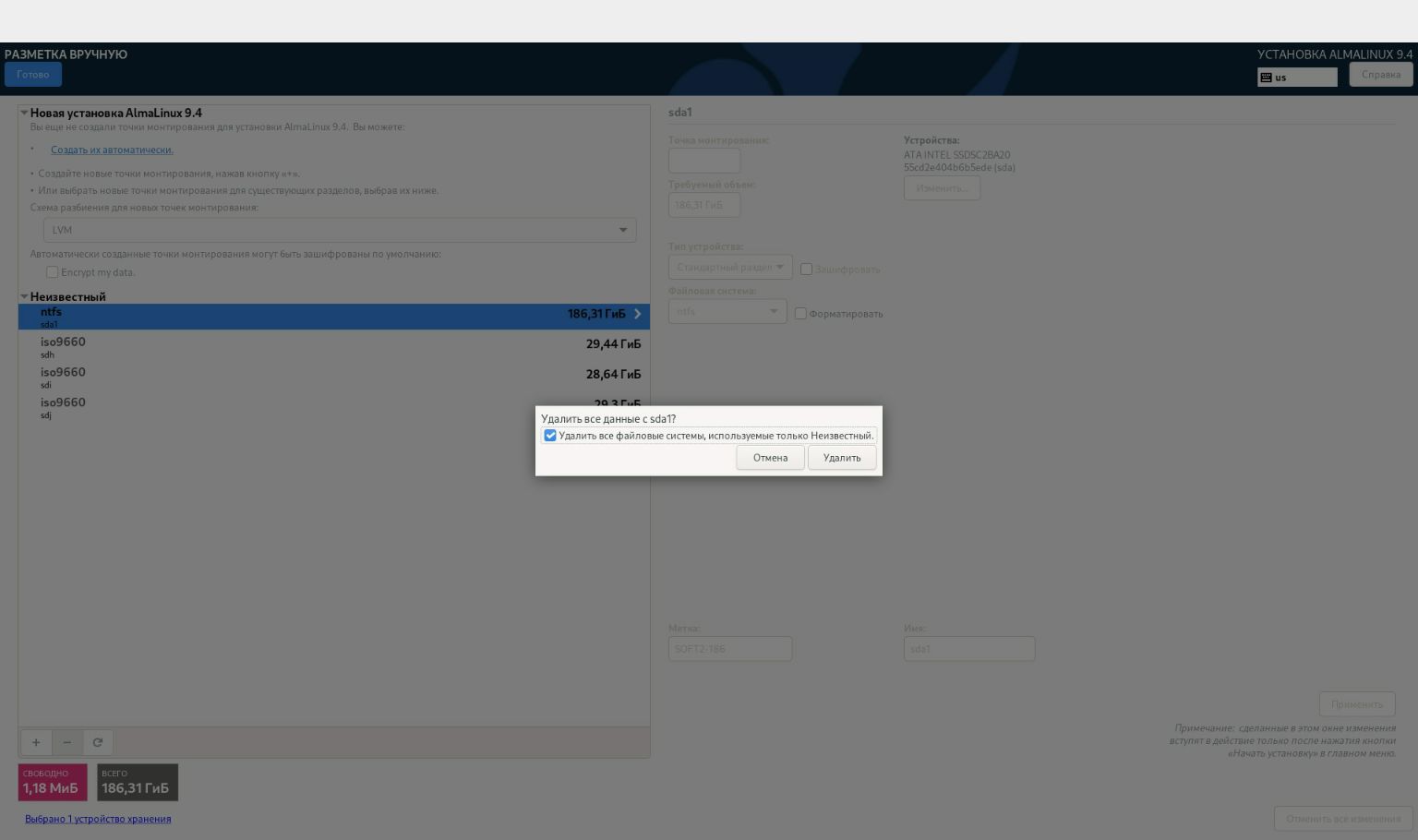
Затем выбираем место установки. Выбираем необходимый диск. Если диска не видно, то встаем в строку дисков и перемещаемся по ней. Выбираем «Конфигурация устройства хранения» «По-своему» и нажимаем «Готово»



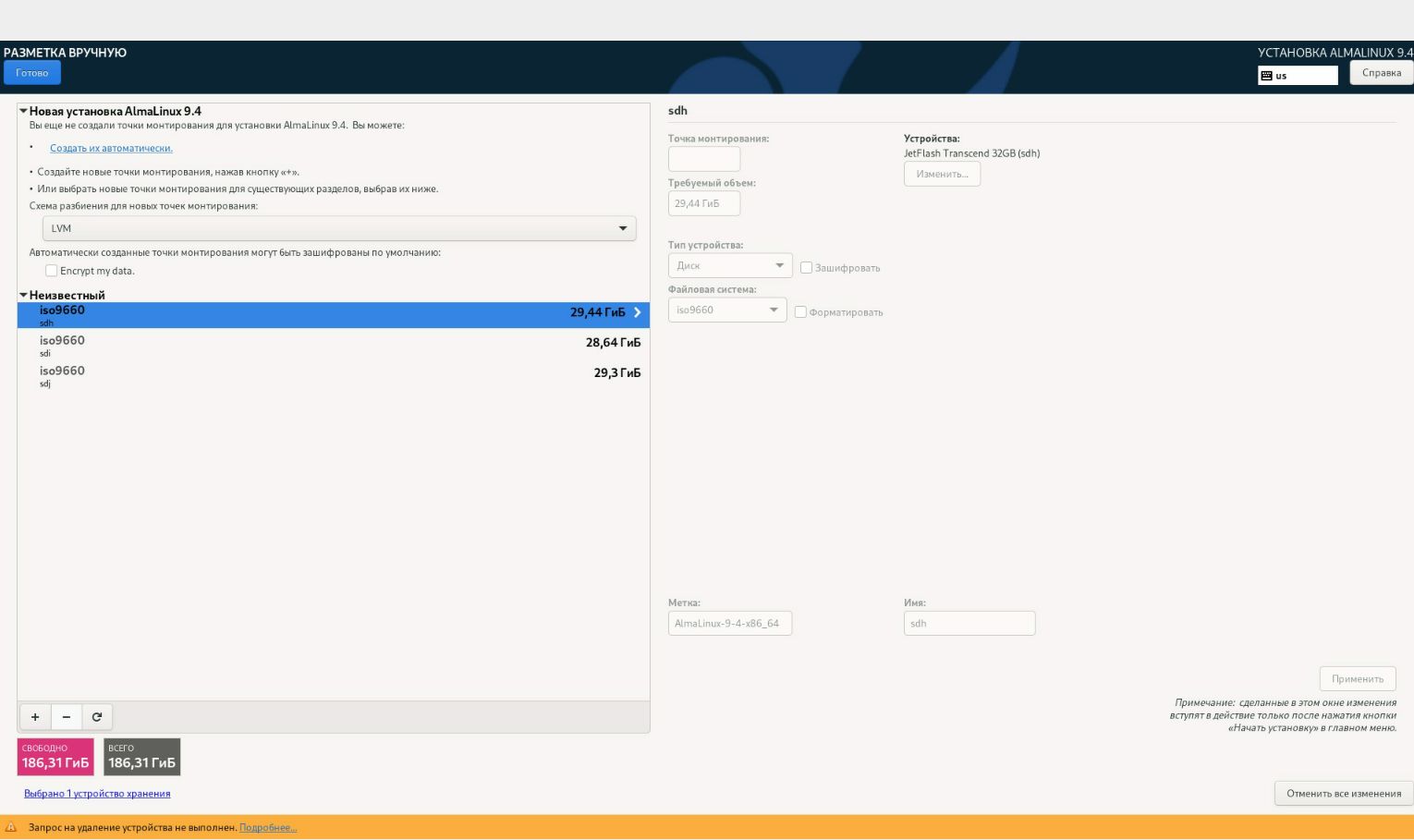
Появляется следующее меню. Так как у меня диск «бэушный», то места на нем нет и его надо освободить.



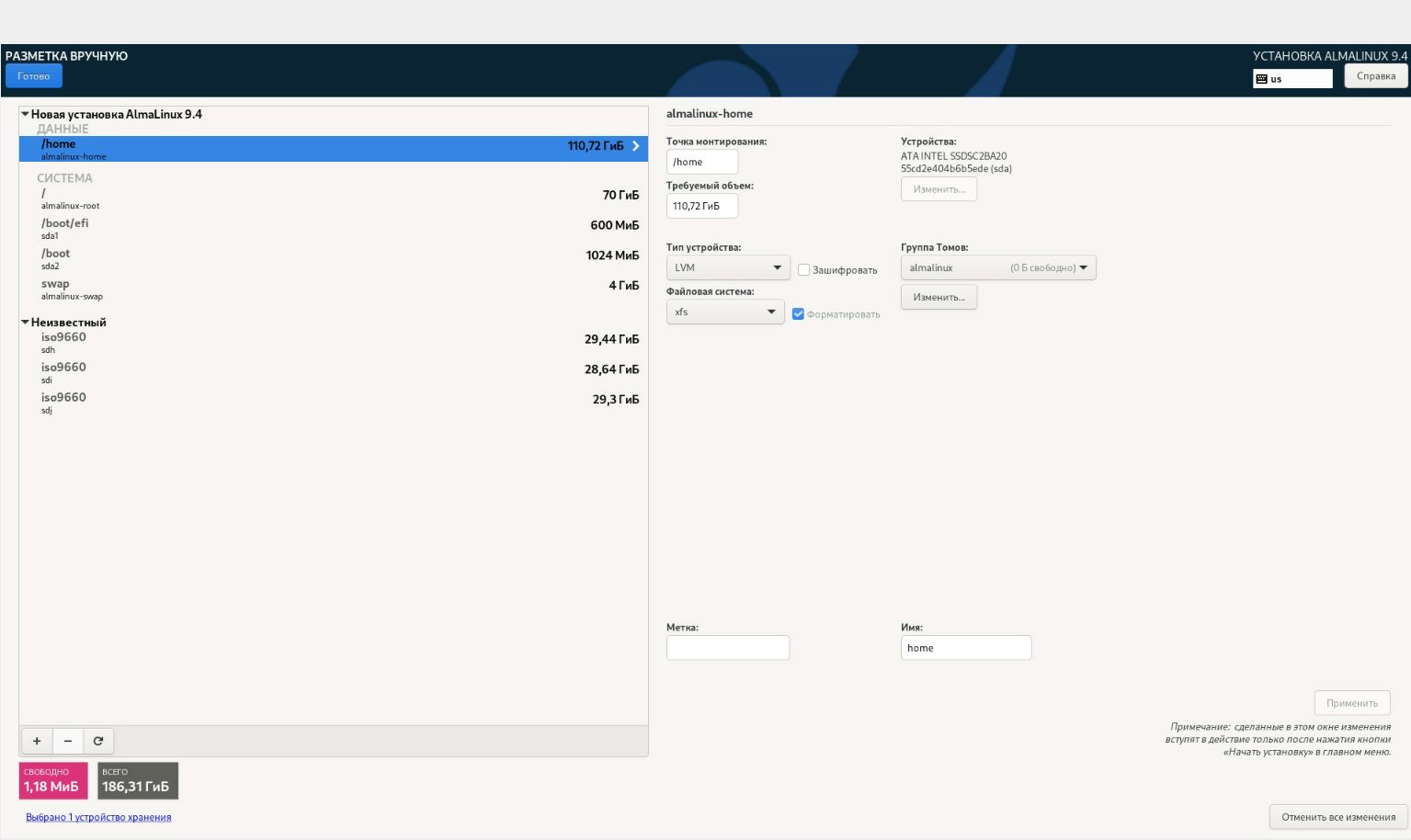
Очищаем наш диск.



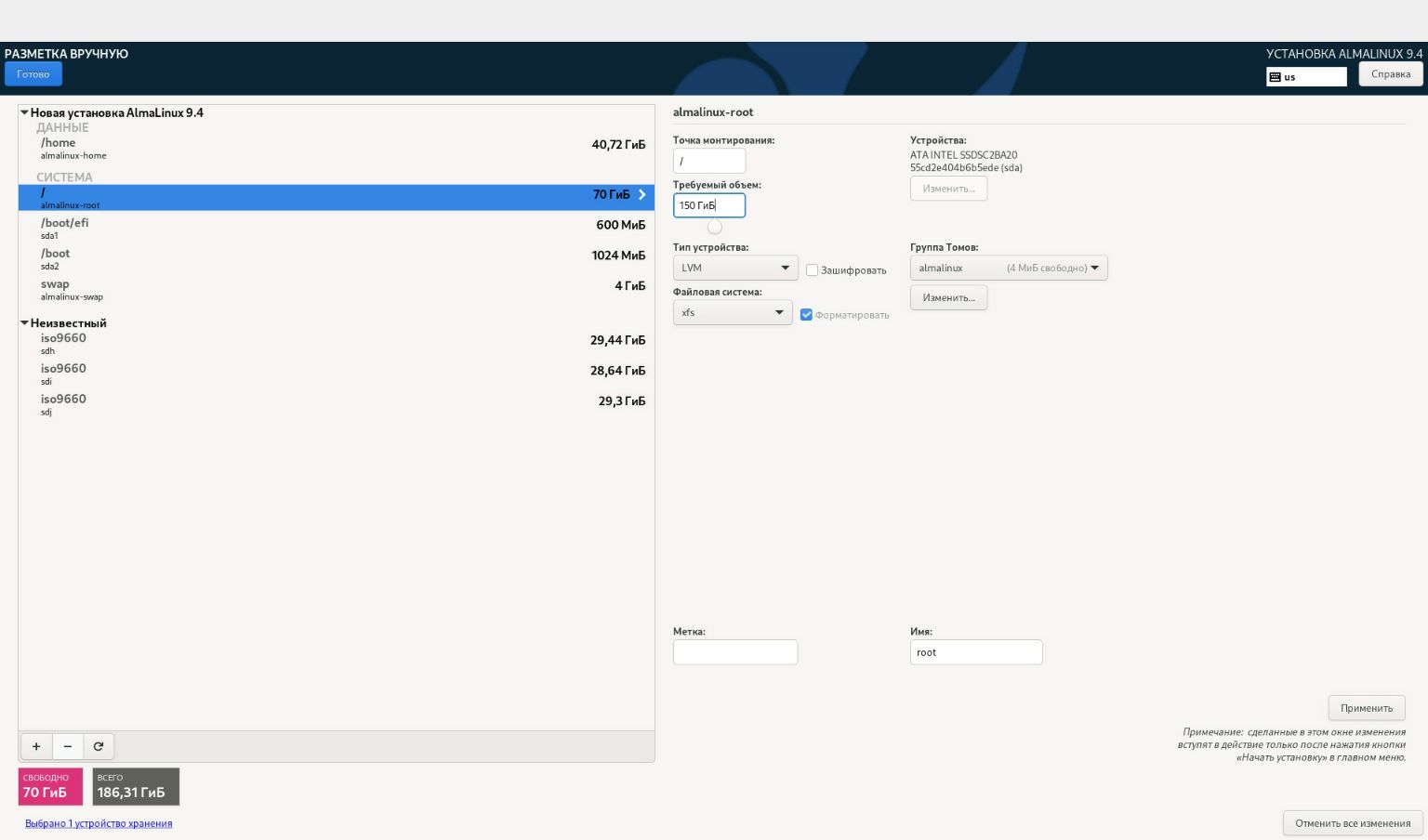
Появляется следующее меню.



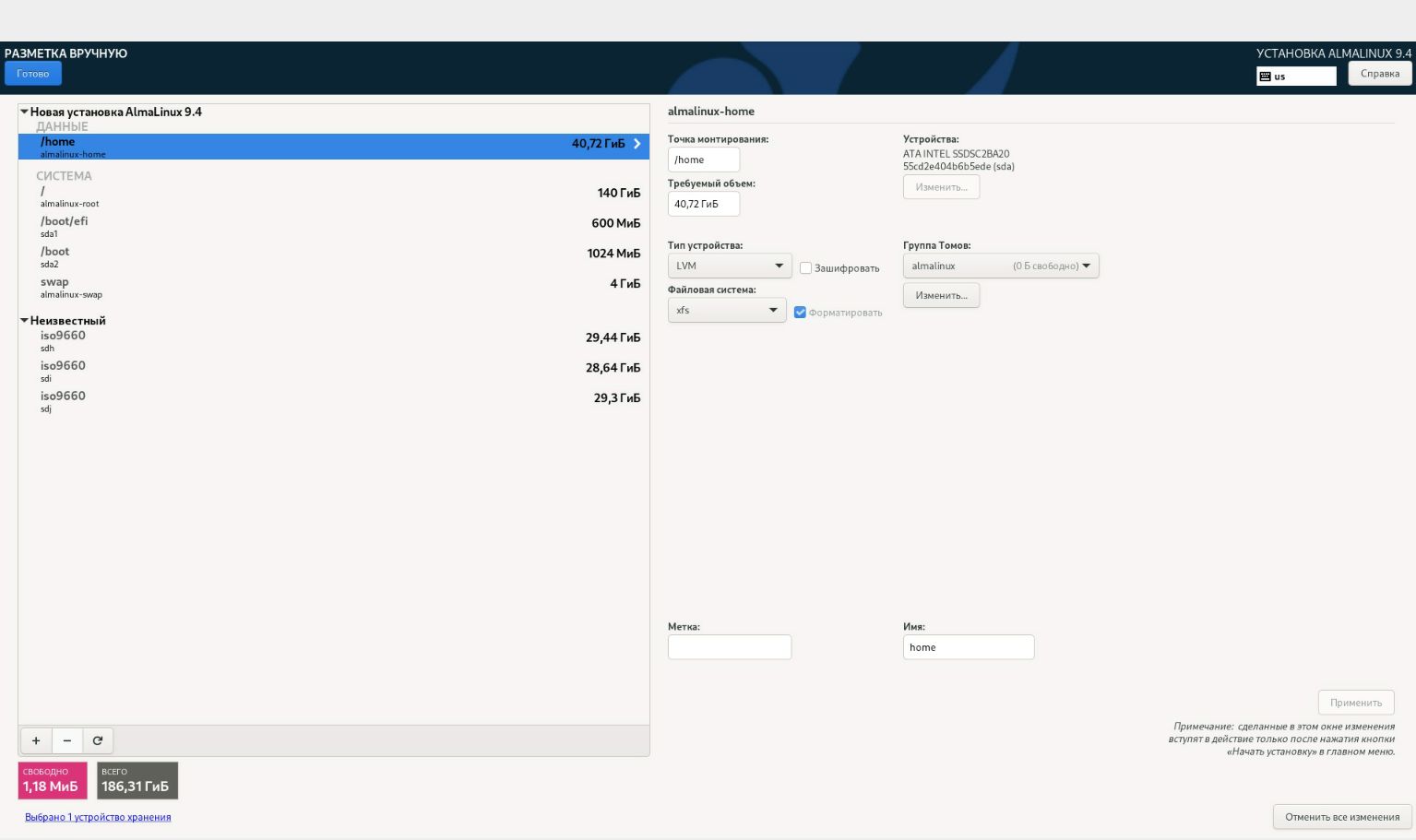
В разделе «Новая установка AlmaLinux 9.4» нажимаем «Создать их автоматически». Получаем следующее



Так-как в дальнейшем мы планируем устанавливать ВМ, а по умолчанию они располагаются в разделе «root», то этот раздел должен иметь необходимый размер. Уменьшаем размер «home», а затем увеличиваем размер «root». Размер можно задать больше свободного. Это будет исправлено автоматически.



В итоге получаем

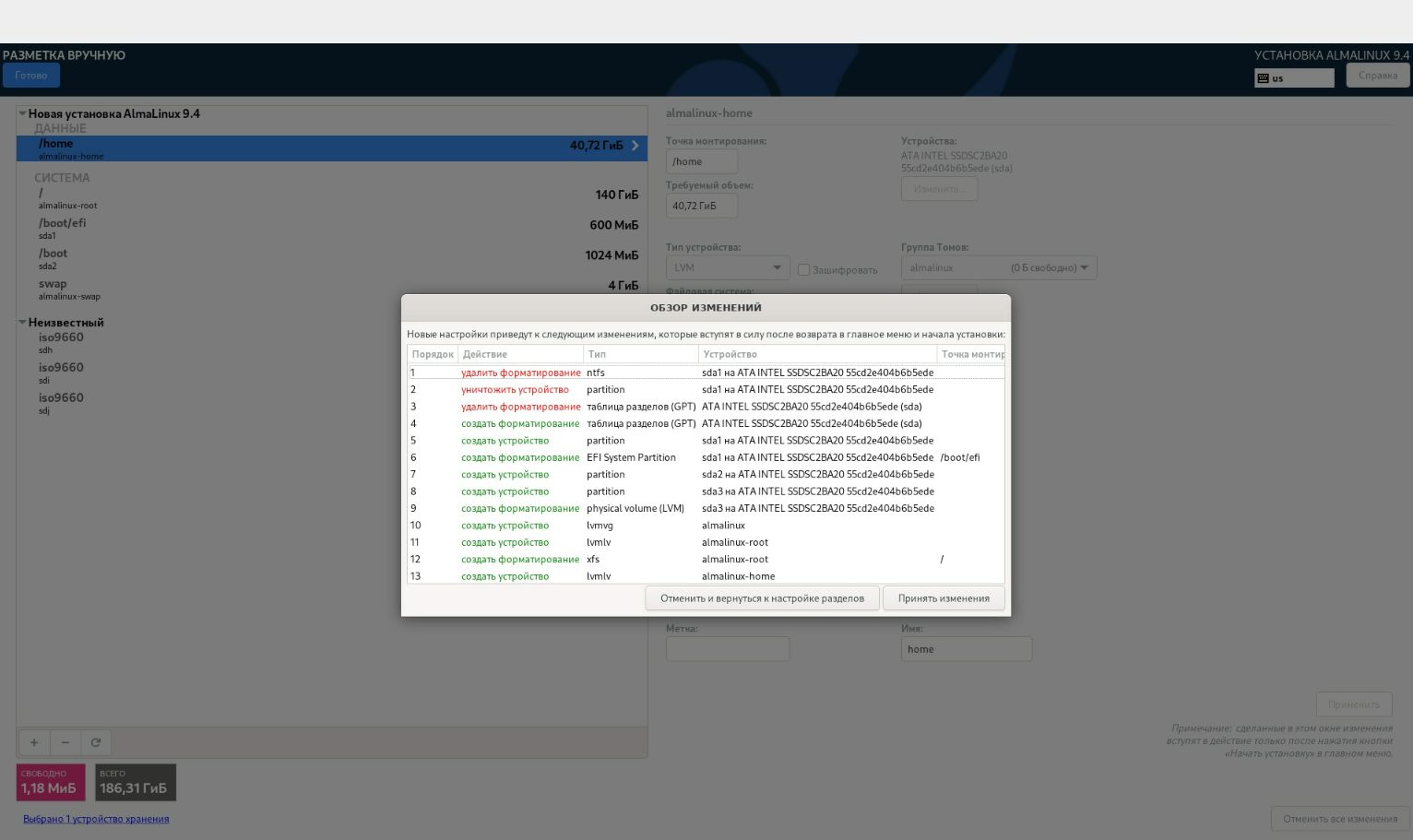


Важно! Так как у меня установлен Raid контроллер, то настройка диска соответствует данной инструкции. Если вы устанавливаете систему без Raid контроллера, то для рабочего сервера надо будет установить не менее двух одинаковых дисков и подготовить их соответствующим образом.

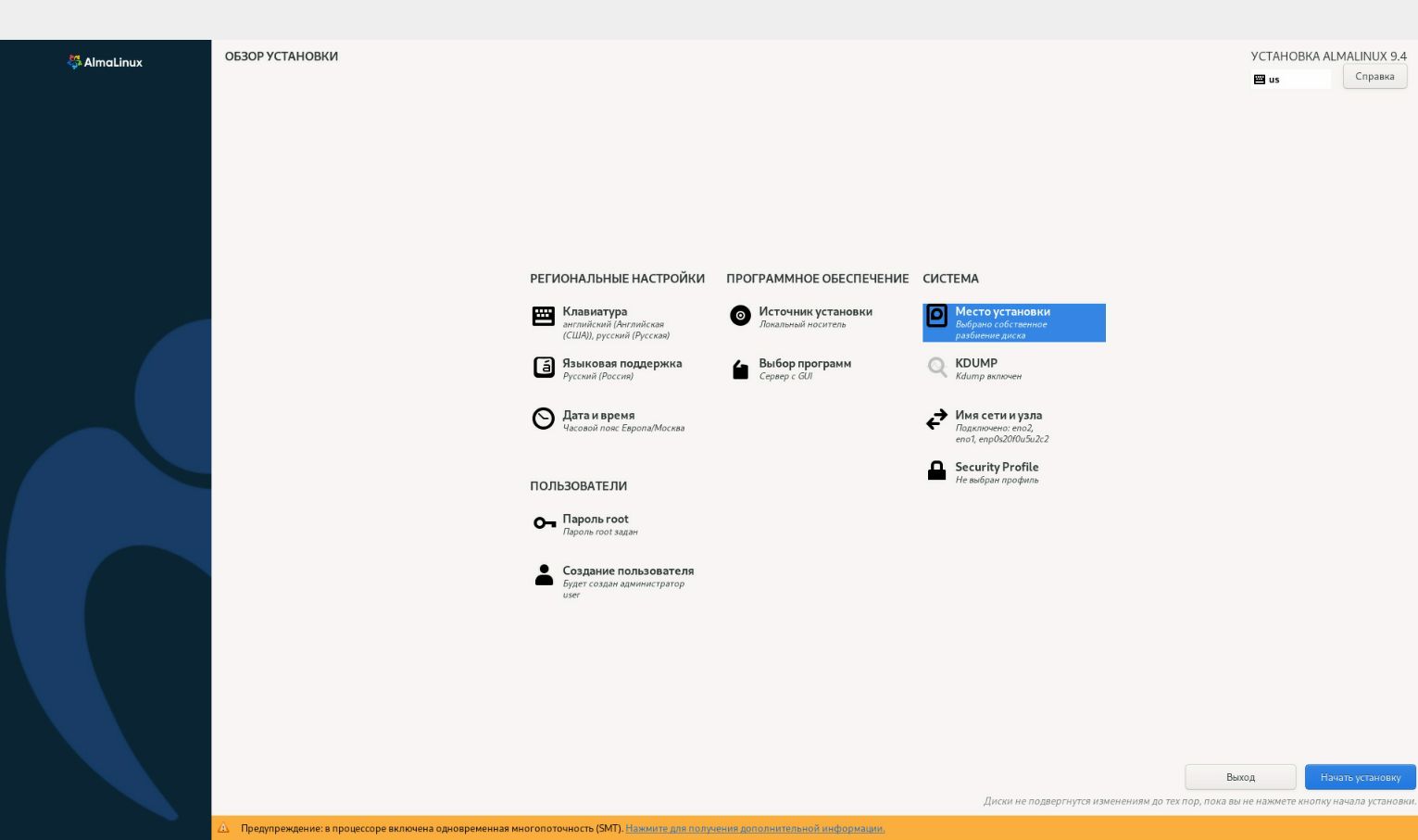
Здесь это не описываю, посмотреть, как это сделать можно на примере установки RED OS по ссылке <https://redos.red-soft.ru/base/redos-8_0/8_0-install/8_0-alter-install/8_0-install-on-raid/8_0-install-on-raid1-and-raid10/?nocache=1730535262817>

И в будущем, если возникают вопросы с работой в AlmaLinux, можете искать инструкции на сайте RED OS

Нажимаем «Готово» и затем «Принять изменения»

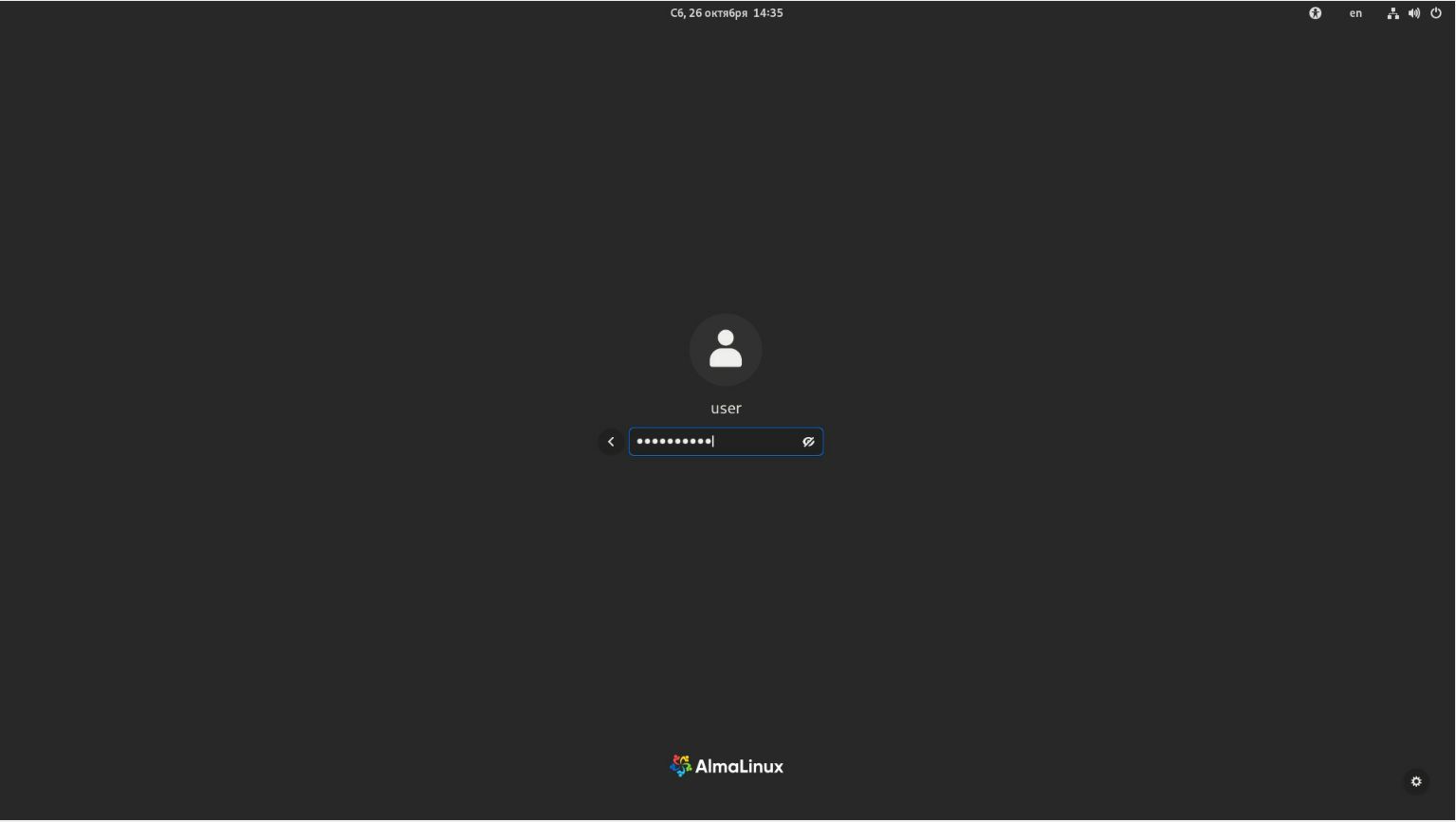


Попадаем в следующее меню. Остальные пункты меню можно пропустить и произвести настройки уже в самой системе после установки. Можно зайти в меню «Имя сети и узла» и посмотреть настройки сети



Начинаем установку.

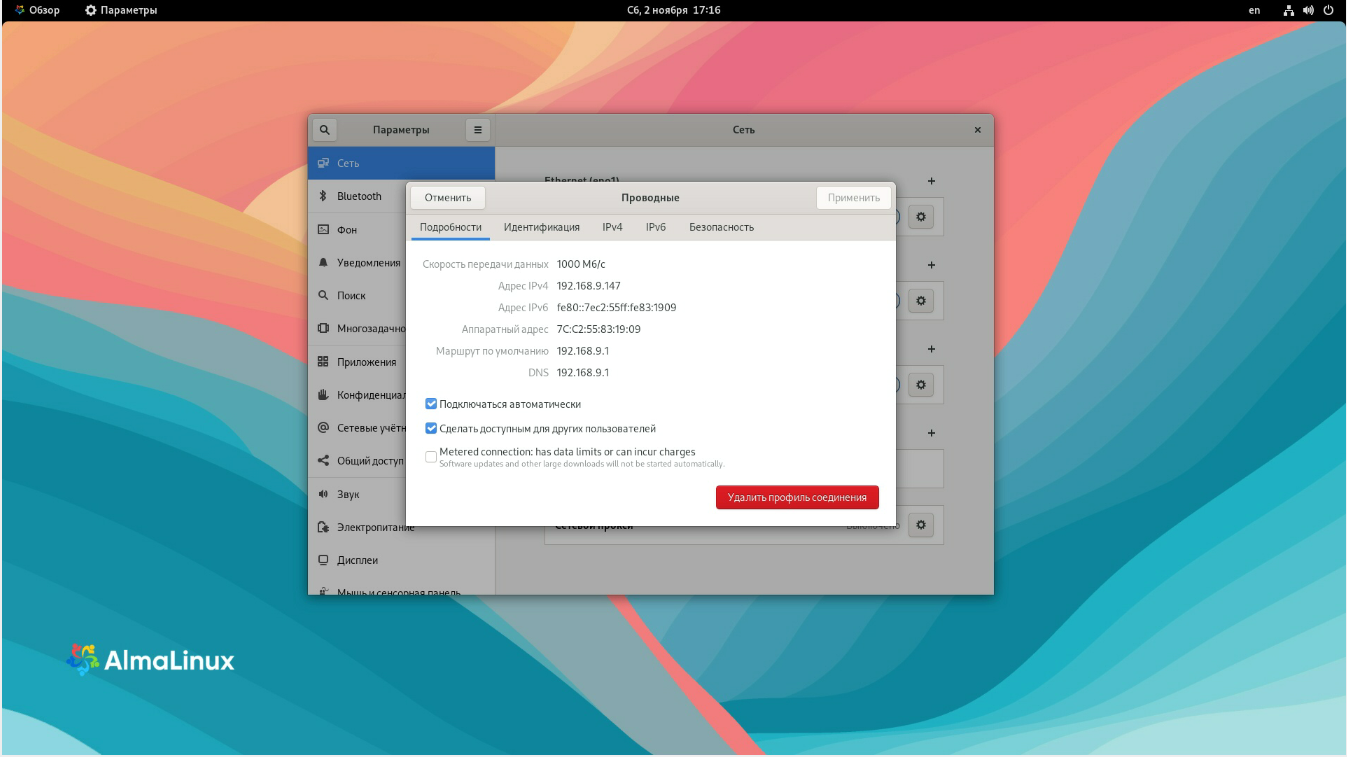
После установки перегружаем сервер и попадаем в окно выбора пользователя



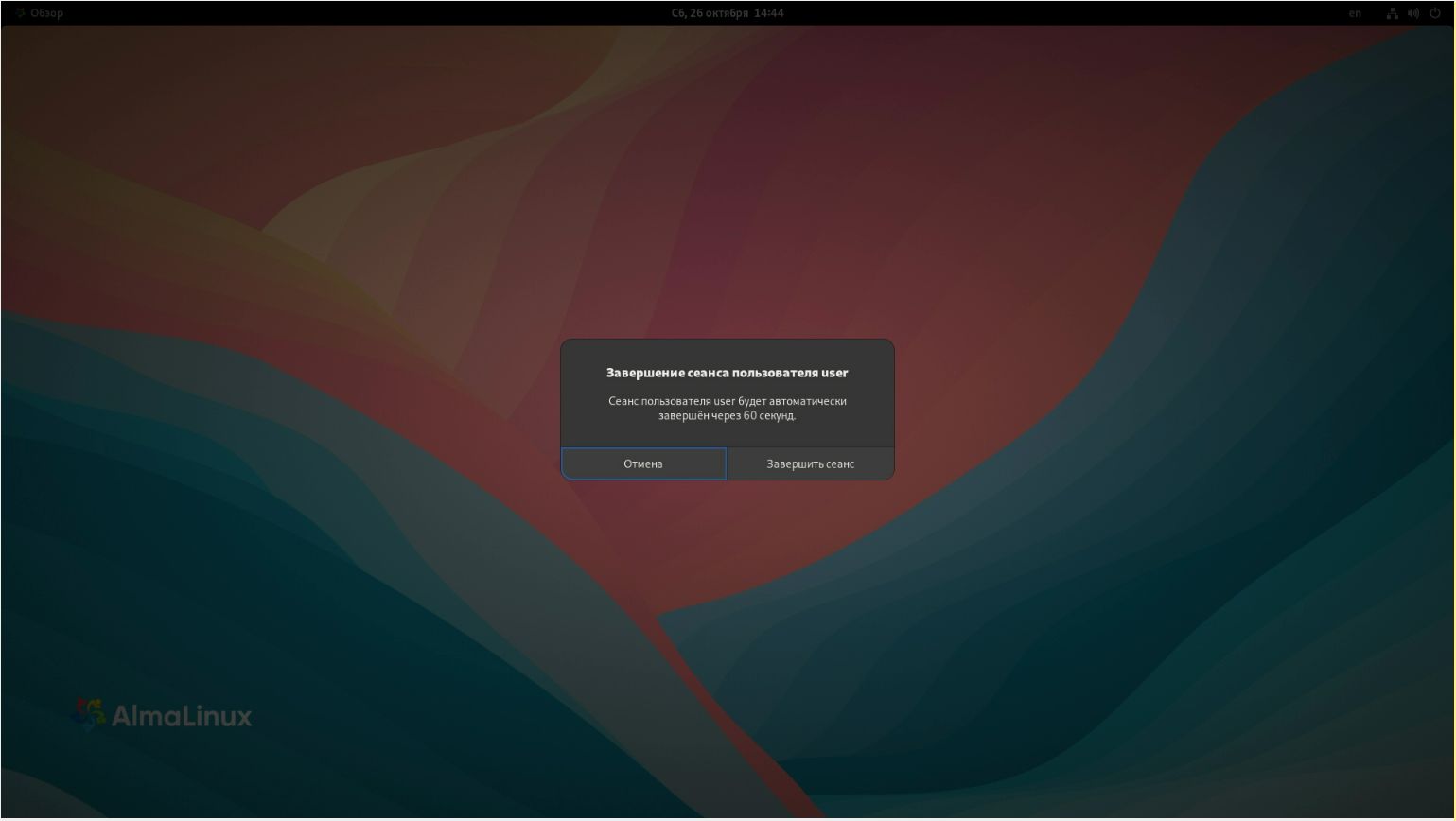


Как работать с линух здесь не описываю. Для работы с сервером в дальнейшем будем пользоваться преимущественно удаленным доступом. Если не знаете IP адрес вашего сервера, то можете в правом верхнем углу открыть меню -> «Параметры» и посмотреть настройки вашей сетевой платы





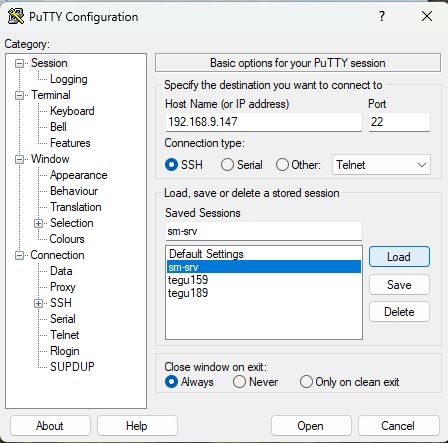
Далее завершаем сеанс пользователя или выключаем сервер



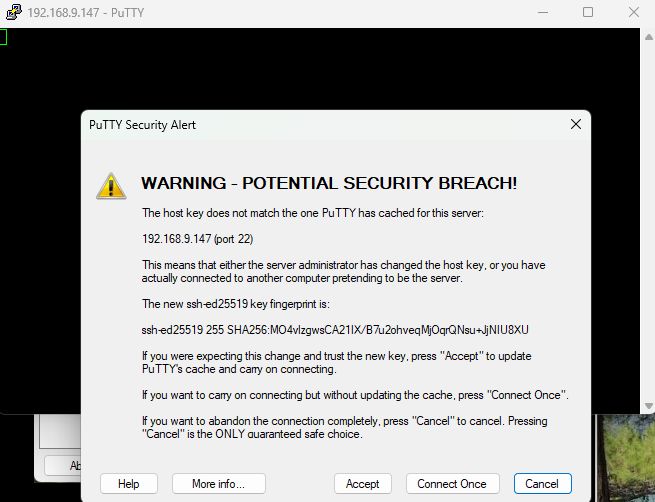
На этом первая часть по подготовке почтового сервера закончена.

Часть вторая. Установка и настройка сервера и клиент Samba.

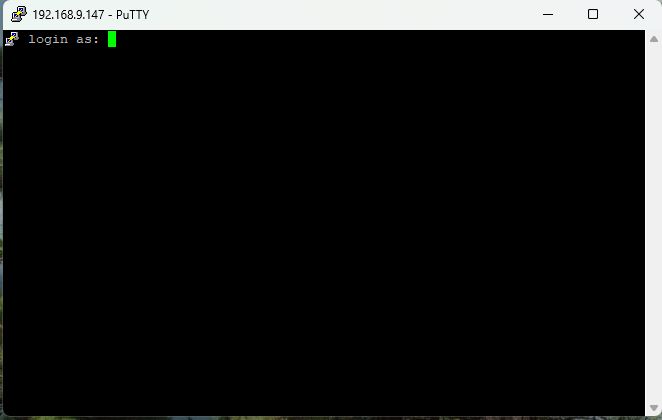
Служба Samba служит для обмена файлами с нашим линукс сервером. Подключаемся к нашему серверу линукс удаленно с любого компьютера windows по протоколу SSH. Это можно сделать через командную строку windows, но мы будем использовать более удобную программу Putty [PuTTY: Telnet/SSH Клиент](https://putty.org.ru/) . Скачиваем, устанавливаем и запускаем Putty.



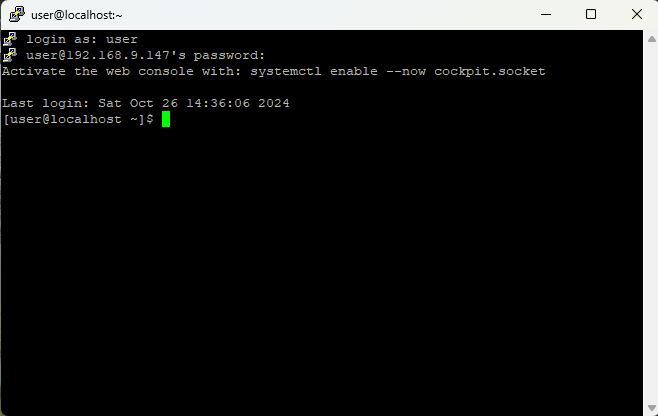
В строке «Host Name (or IP address)» набираем IP адрес нашего сервера, у меня 192.168.9.147, порт оставляем 22. Настройки как на картинке. В строке «Saved Sessions» пишем название нашего подключения и нажимаем «Save». Нажимаем «Open». Если все прописали правильно, то запускается терминальная сессия и должно появиться следующее окно



Нажимаем «Accept», появляется окно запроса логина



Вводим логин нашего пользователя «user», нажимаем «Enter». Появляется запрос на ввод пароля. Вводим пароль (при вводе пароль не отображается) и «Enter».

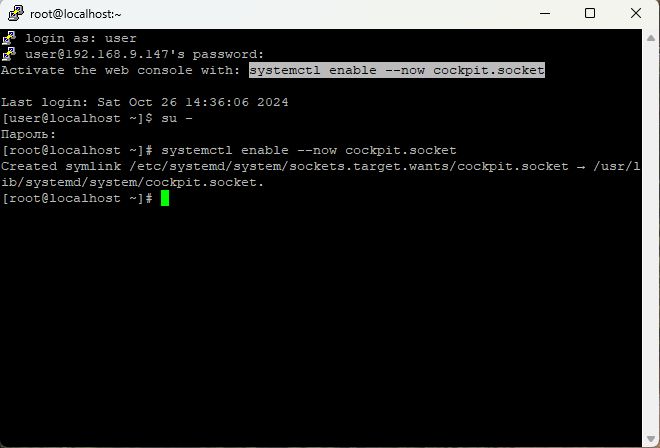


**В Linux для всех серьёзных настроек или операций с системой нужны права суперпользователя (root). Пользователям Windows такой подход знаком, когда вы устанавливаете программы или меняете системные настройки, операционная система просит подтвердить выполнение программы от имени администратора. В Linux такой возможности нет, но зато есть команда «sudo», которая позволяет вам запускать программы от имени других пользователей, а также от имени суперпользователя. В Almalinux, чтобы иметь возможность использовать команду sudo, пользователь должен входить в группу wheel. При установке системы наш «user» был добавлен в эту группу (группа администраторов).**

Так как при установке AlmaLinux 9 автоматически устанавливаются компоненты Cockpit, мы видим приглашение по его активации. Для активации переходим или в сеанс root следующей командой «**su –**» и на запрос вводим пароль пользователя root. Затем выполняем команду активации Cockpit. Или перед командой вставляем «sudo», команда будет выглядеть так:

**sudo systemctl enable --now cockpit.socket**

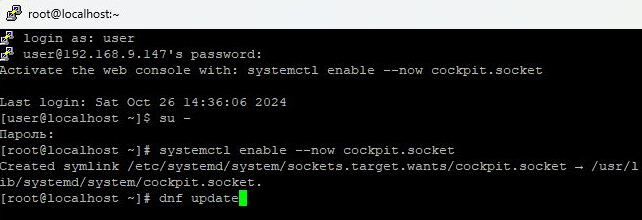
И на запрос пароля вводим пароль пользователя «user»



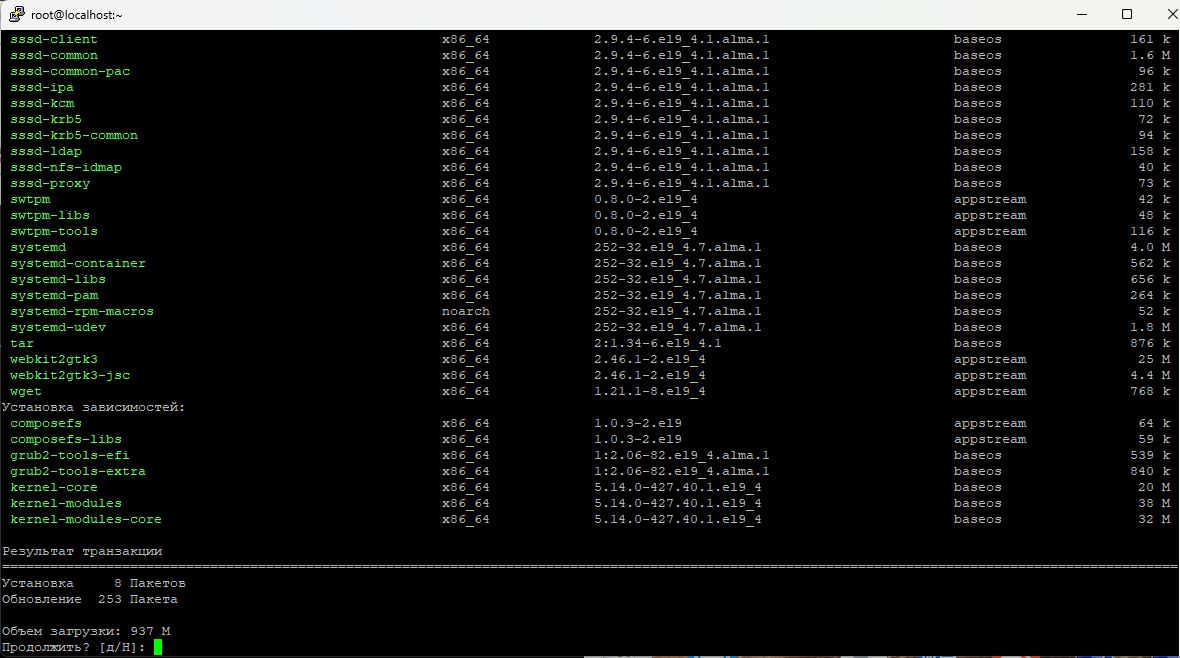
По работе с Cockpit вернемся в конце этой главы.

Переходим к установке Samba. Перед установкой каких-то программ необходимо проверить наличие обновлении нашего линукс. Это делается командой

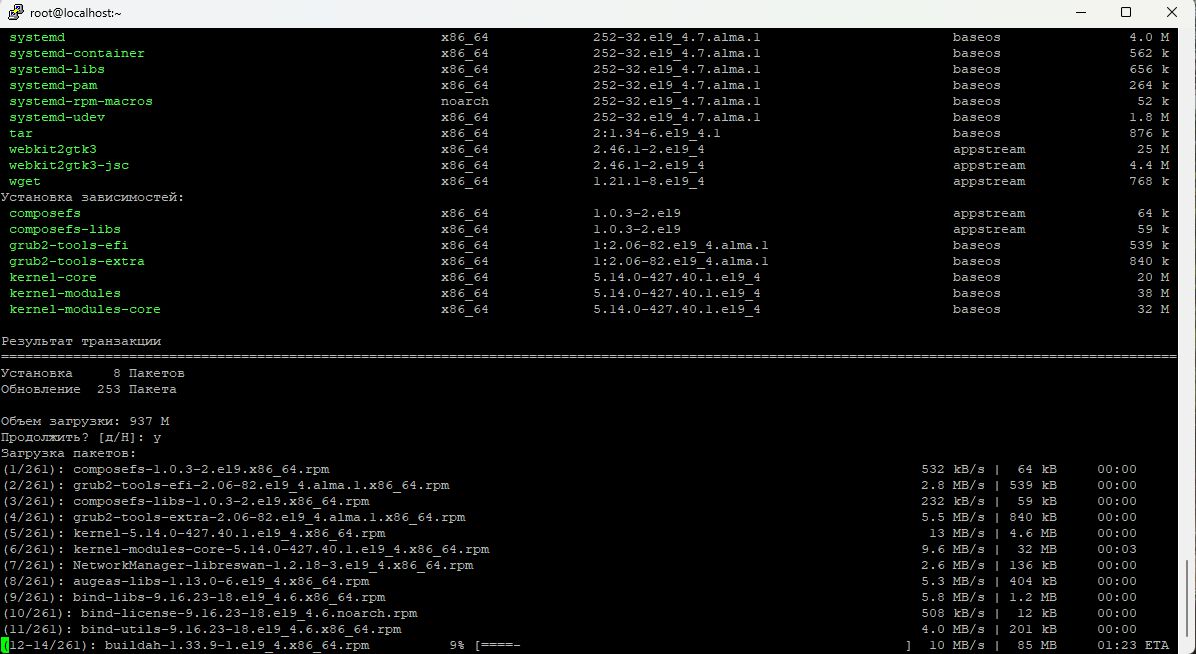
**dnf update**



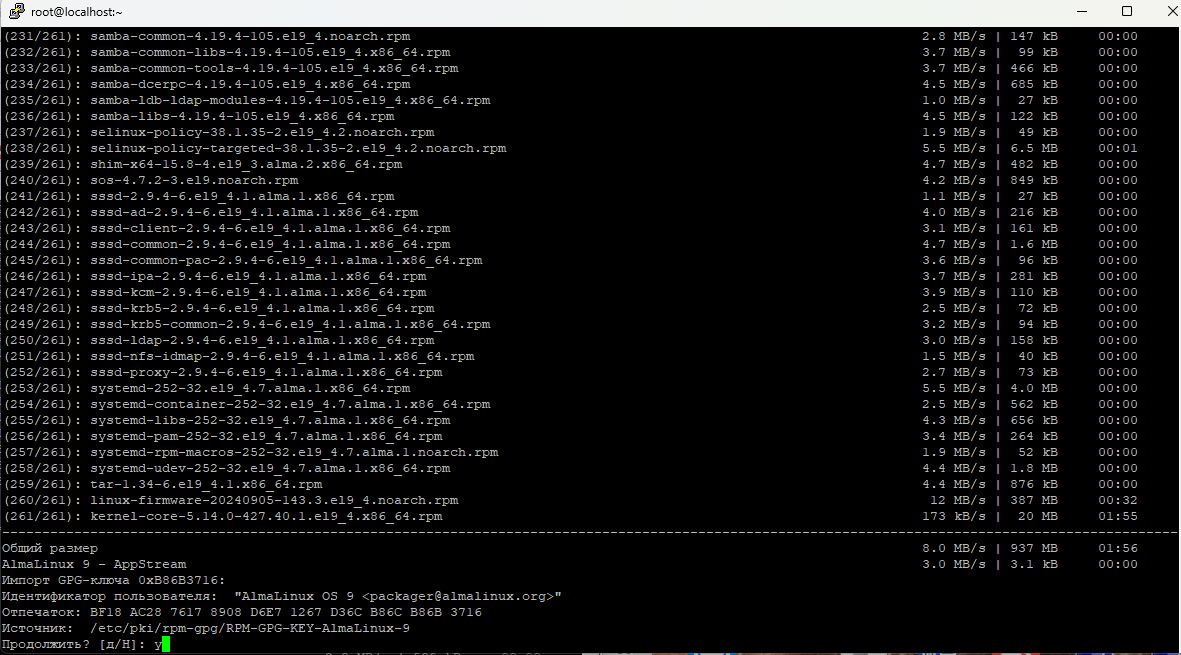
Выполняем команду и на запрос продолжения обновления набираем «y»



Начинается процесс загрузки обновлений



После загрузки всех обновлений появится запрос на их установку. Подтверждаем



Начинается процесс установки обновлений. Это займет некоторое время.

Устанавливаем репозиторий epel-release и снова проверяем обновления

**dnf install epel-release**

**dnf update -y**

Теперь непосредственно переходим к установке и настройке Samba

Так-как у нас пакеты уже установлены, то на некоторые команды будет выводится сообщение об это. Ничего страшного. Последовательно выполняем выделенные команды

Установка Samba

**dnf install samba samba-client**

Запускаем сервис Samba и включаем его в автозагрузку

**systemctl enable --now {smb,nmb}**

Разрешение доступ к ресурсам Samba через firewall

**firewall-cmd --permanent --add-service=samba**

**firewall-cmd --reload**

Две команды ниже мы не будем выполнять. Приведены для общей информации.

По умолчанию в настройках Samba на Almalinux 9 доступ к ресурсам Samba не ограничен какими-то группами.

Создание пользователя Samba

adduser -M sambauser -s /sbin/nologin

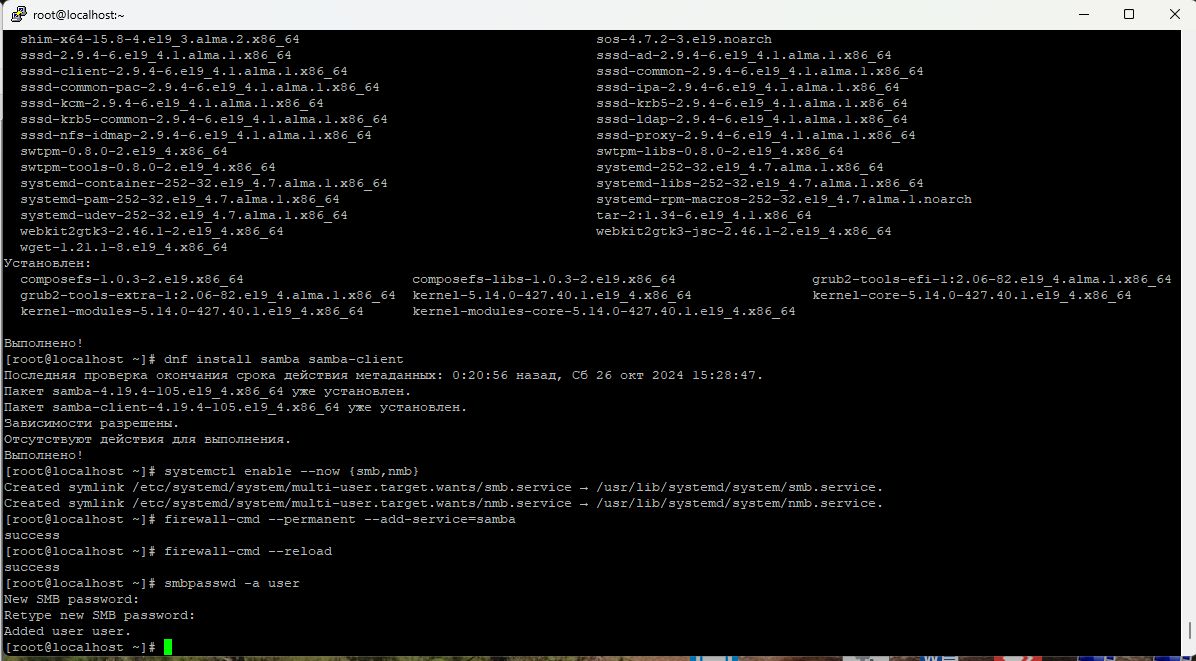
Добавление существующего пользователя в группу samba

usermod -aG smbshare sambauser

Мы не будем создавать специального пользователя Samba. В качестве такого пользователя будем использовать нашего пользователя «user».

Создаем пароль Samba для пользователя (чтобы не путаться оставляем существующий пароль)

**smbpasswd -a user**

****

Далее, для привычного пользователям windows обмена файлами, создаем папку /home/distrib и настраиваем Samba для доступа к ней (ключ -p не выдавать запрос если папка существует)

**mkdir -p /home/distrib**

Меняем группу доступа к этой папке на пользователя "user"

**chgrp -R user /home/distrib**

Даем все права для этой папки

**chmod 777 /home/distrib**

Проверяем права

**ls -l /home**

Должно получится

итого 4

drwxrwxrwx. 2 root user 6 окт 26 16:00 distrib

drwx------. 14 user user 4096 окт 26 14:36 user

Настраиваем конфигурационный файл Samba (редактором nano)

**nano /etc/samba/smb.conf**

Открывается файл в редакторе nano. По файлу перемещаемся с помощью стрелок на клавиатуре. Добавляем в конец файла

**[distrib]**

**path = /home/distrib**

**guest ok = no**

**read only = no**

Сохраняем файл нажимая Ctrl+O затем "Enter". Выход из редактора Ctrl+X

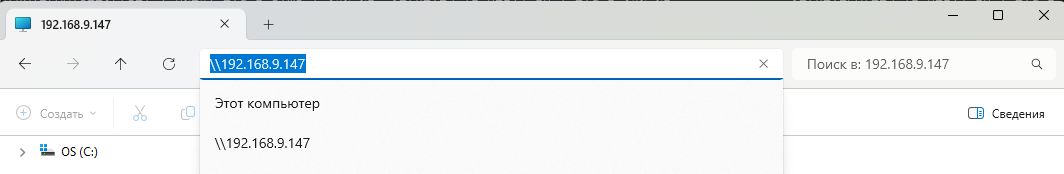
Перезапускаем сервис Samba

**systemctl restart {smb,nmb}**

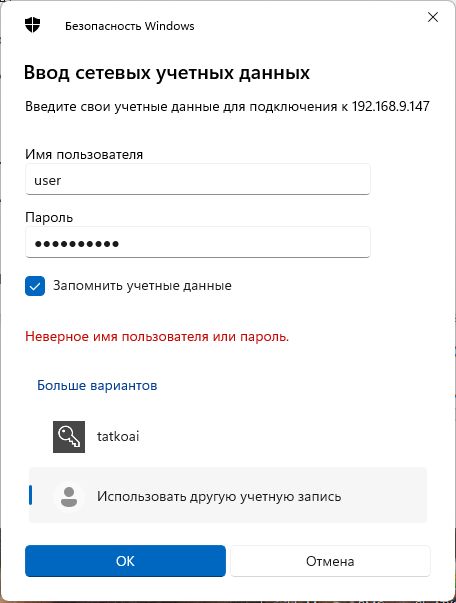
Настройка политики SELinux для доступа к папке Samba

**chcon -R -t samba\_share\_t /home/distrib**

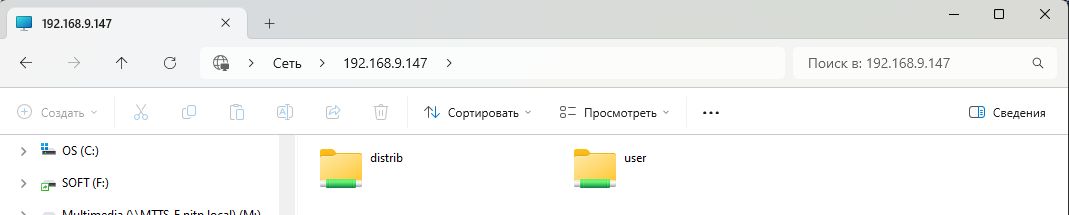
Открываем проводник Windows и вводим адрес нашего сервера (замените на адрес своего сервера) [\\192.168.9.147](file:///\\192.168.9.147) Нажимаем «Enter»



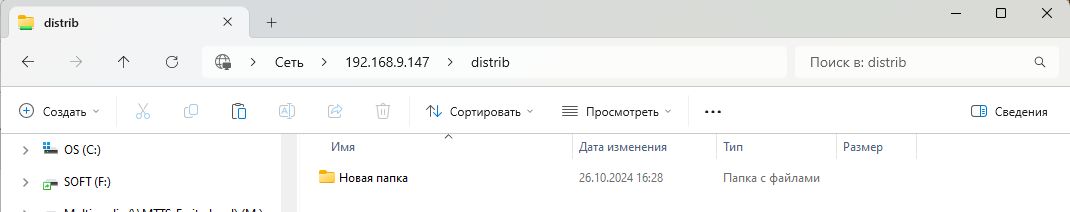
Появляется запрос на ввод пароля. Вводим имя пользователя Samba и пароль.



Открывается следующее окно.



Открываем папку distrib и, для проверки создаем внутри папку. Папка создалась, значит все получилось. Для будущих установок ВМ копируем в эту папку iso образ установочного диска Almalinux 9. На этом настройка сервера Samba закончена.



Для пользователей windows есть удобная программа с привычным графическим интерфейсом для удаленно работы с системами линукс. Она может использовать сохраненные настройки подключения из Putty. В этой инструкции описывать ее не буду. Получить информацию по программе и скачать ее можно по ссылке <https://winscp.net/eng/docs/lang:ru>

Но надо иметь ввиду, что для ее полноценного использования нужно заходит с правами пользователя «root». Будьте внимательны, можете нечаянно повредить систему.

Теперь переходим к настройку Cockpit.

Cockpit — это удобный и простой в освоении инструмент, предназначенный для мониторинга и администрирования серверов под управлением Linux через веб-браузер.

Авторизация в веб-интерфейсе этого инструмента происходит с использованием данных локальной учетной записи Вашей системы.

Права доступа, которыми обладает выбранная учетная запись, определяют доступные Cockpit возможности по управлению системой. То есть для выполнения большинства операций нужно авторизоваться с помощью данных аккаунта, имеющего права администратора.

Системы под управлением дистрибутивов Almalinux 9 уже содержат установленный инструмент Cockpit. Если приложение отсутствует, его можно установить.

**dnf install cockpit cockpit-machines**

Далее нужно дождаться установки указанного пакета, после чего можно активировать сокет Cockpit

**systemctl start cockpit.socket**

Также включаем его автозапуск после включения системы:

**systemctl enable cockpit.socket**

Проверяем статус сервиса

**systemctl status cockpit.socket**

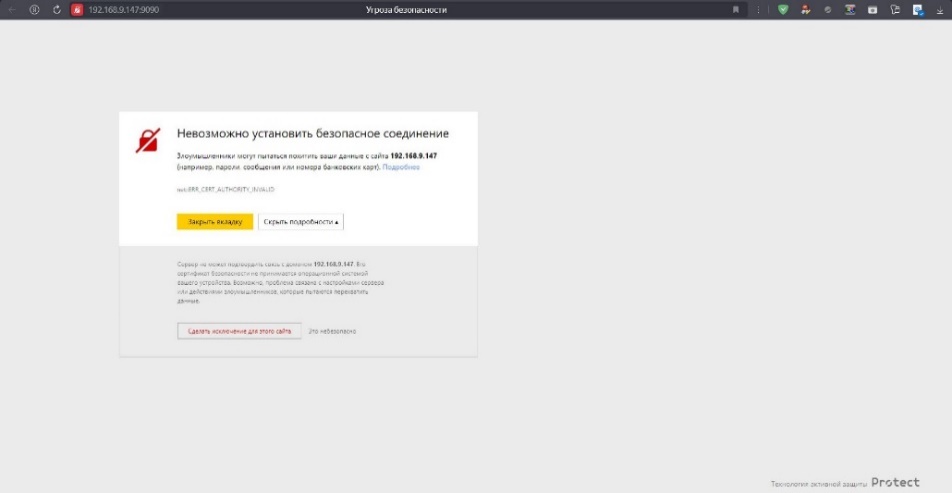
Далее добавляем службу Cockpit в исключения брандмауэра и перезагрузите его:

**firewall-cmd --add-service=cockpit --permanent**

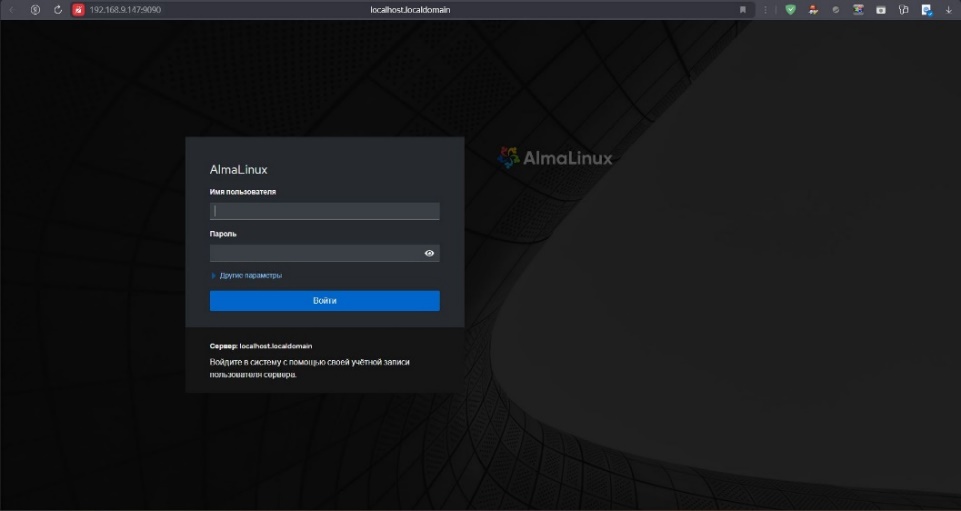
**firewall-cmd --reload**

Теперь запускаем веб-браузер и вводим адрес вашего сервера:

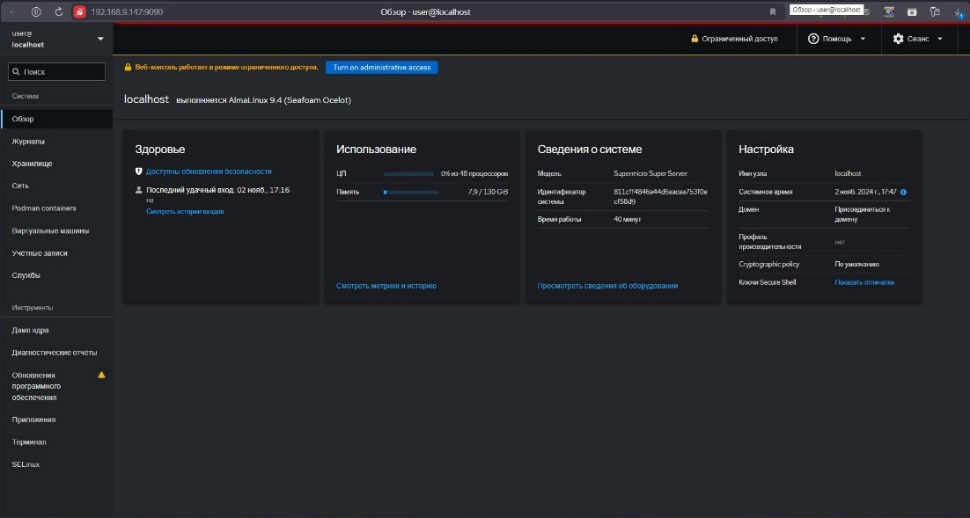
<https://SERVER_IP:9090/>



В результате должен открыться веб-интерфейс Cockpit. Стоит отметить, что данный инструмент использует самозаверенный сертификат SSL для включения HTTPS. То есть, для доступа к нему нужно в ответ на предупреждение от браузера подтвердить свои действия. В результате увидим окно авторизации, куда вводим данные вашей учетной записи Linux.



Если Вы правильно ввели свои данные и правильно выполнили предыдущие шаги, то откроется веб-консоль приложения.



Работу в Cockpit здесь не описываю. Изучайте самостоятельно.

Чтобы завершить сеанс терминала выполняем команду «**exit**». Если вы в терминале открывали сеанс пользователя root командой «**su –**», то эту команду надо будет повторить два раза.

Часть третья. Удаленный рабочий стол Xrdp.

Подключаемся к серверу по SSH при помощи Putty переключаемся на суперпользователя root.

Графическая оболочка GNOME у нас была установлена при начальной установке сервера.

Если нет, то устанавливаем

dnf groupinstall -y "Server with GUI"

systemctl set-default graphical.target

systemctl default

Устанавливаем epel-release и проверяем обновления

**dnf install epel-release**

**dnf update -y**

Устанавливаем Xrdp

**dnf install xrdp**

Запускаем Xrdp и включаем его в автозагрузку.

**systemctl start xrdp**

**systemctl enable xrdp**

Настраиваем Firewall для разрешения подключения по RDP

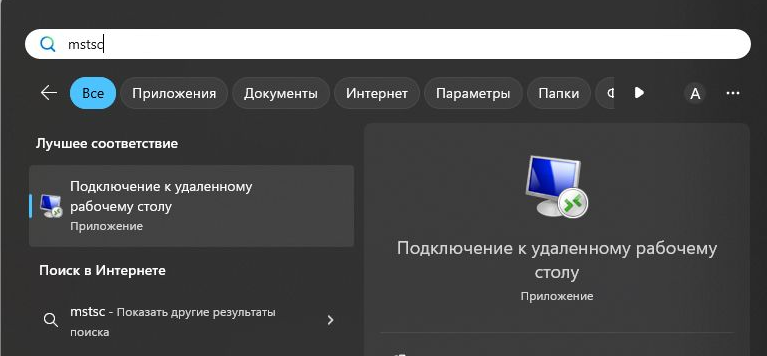
**firewall-cmd --permanent --add-port=3389/tcp**

**firewall-cmd --reload**

Как правило в сессии RDP из под Windows не работает переключение на русский язык

Ниже мы это исправим.

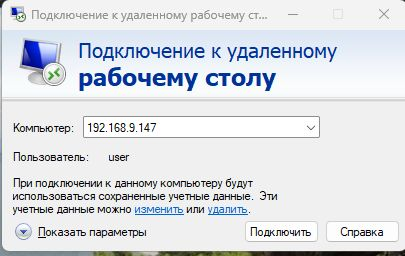
Можете это проверить самостоятельно запустив из под Windows сеанс "Подключение к удаленному рабочему столу". Кто не знает что это такое, то набираем в поиске mstsc и нажимаем на соответствующий значок.

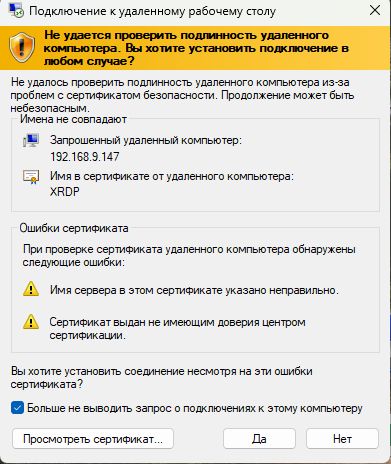


Перед подключением выбираем раскладку ENG

Во время регистрации изменить раскладку клавиатуры нельзя.

По умолчанию переключение языка в Almalinux клавищей "Super" + "Пробел" на клавиатуре Windows клавиша "Super" это клавиша "Windows"









Исправление переключения русского языка

Устанавливаем xorgxrdp

**dnf install xorgxrdp**

Редактируем /etc/xrdp/xrdp.ini

**nano /etc/xrdp/xrdp.ini**

Нужно найти и раскоментировать [Xorg] блок в xrdp.ini

Должно получиться так

[Xorg]

name=Xorg

lib=libxup.so

username=ask

password=ask

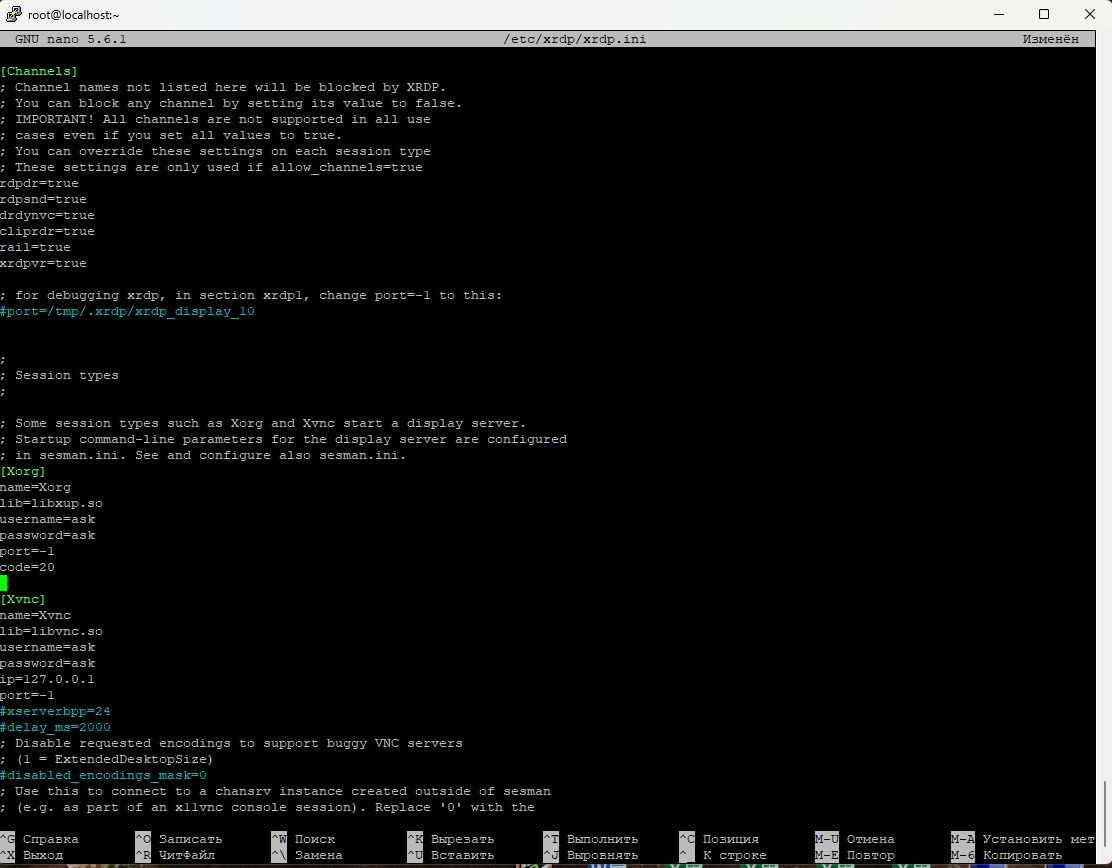
port=-1

code=20

В блоке [Xvnc] ничего не меняем

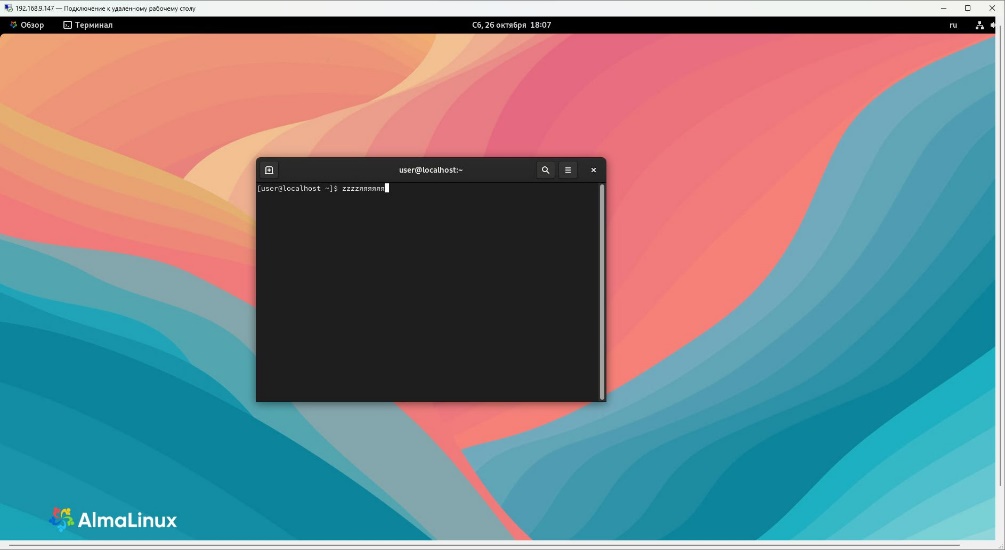
Сохраняем файл и перегружаем xrdp service

**sudo systemctl restart xrdp**

****

Теперь переключение на русский язык в сессии RDP из Windows должно работать.

Подключаемся к серверу (см. рисунки выше). Запускаем терминал (черный квадрат на панели внизу экрана). И проверяем переключение языка (переключение «Windows» + «Пробел»



Часть четвертая. Установка ядра 6.Х на Almalinux9.

Проверяем текущую версию ядра

**uname -sr**

Получаем примерно так

[root@localhost ~]# uname -sr

Linux 5.14.0-427.40.1.el9\_4.x86\_64

Подключаем репозиторий elrepo

**rpm --import https://www.elrepo.org/RPM-GPG-KEY-elrepo.org**

**dnf install https://www.elrepo.org/elrepo-release-9.el9.elrepo.noarch.rpm**

Получаем

elrepo-release-9.el9.elrepo.noarch.rpm 6.2 kB/s | 12 kB 00:02

Зависимости разрешены.

=====================================================================================

Пакет Архитектура Версия Репозиторий Размер

=====================================================================================

Установка:

elrepo-release noarch 9.1-1.el9.elrepo @commandline 12 k

Результат транзакции

=====================================================================================

Установка 1 Пакет

Общий размер: 12 k

Объем изменений: 5.0 k

Продолжить? [д/Н]: y

Загрузка пакетов:

Проверка транзакции

Проверка транзакции успешно завершена.

Идет проверка транзакции

Тест транзакции проведен успешно.

Выполнение транзакции

Подготовка : 1/1

Установка : elrepo-release-9.1-1.el9.elrepo.noarch 1/1

Проверка : elrepo-release-9.1-1.el9.elrepo.noarch 1/1

Установлен:

elrepo-release-9.1-1.el9.elrepo.noarch

Выполнено!

Устанавливаем ядро 6 версии

**dnf --enablerepo=elrepo-kernel install kernel-ml**

Получаем

ELRepo.org Community Enterprise Linux Repository - el9 116 kB/s | 291 kB 00:02

ELRepo.org Community Enterprise Linux Kernel Repository - el9 3.3 MB/s | 3.1 MB 00:00

Зависимости разрешены.

=====================================================================================

Пакет Архитектура Версия Репозиторий Размер

=====================================================================================

Установка:

kernel-ml x86\_64 6.11.5-1.el9.elrepo elrepo-kernel 46 k

Установка зависимостей:

kernel-ml-core x86\_64 6.11.5-1.el9.elrepo elrepo-kernel 53 M

kernel-ml-modules x86\_64 6.11.5-1.el9.elrepo elrepo-kernel 51 M

Результат транзакции

=====================================================================================

Установка 3 Пакета

Объем загрузки: 104 M

Объем изменений: 150 M

Продолжить? [д/Н]: y

Загрузка пакетов:

(1/3): kernel-ml-6.11.5-1.el9.elrepo.x86\_64.rpm 233 kB/s | 46 kB 00:00

(2/3): kernel-ml-modules-6.11.5-1.el9.elrepo.x86\_64.rpm 19 MB/s | 51 MB 00:02

(3/3): kernel-ml-core-6.11.5-1.el9.elrepo.x86\_64.rpm 11 MB/s | 53 MB 00:05

----------------------------------------------------------------------------------------------------------------------

Общий размер 9.9 MB/s | 104 MB 00:10

Проверка транзакции

Проверка транзакции успешно завершена.

Идет проверка транзакции

Тест транзакции проведен успешно.

Выполнение транзакции

Подготовка : 1/1

Установка : kernel-ml-core-6.11.5-1.el9.elrepo.x86\_64 1/3

Запуск скриптлета: kernel-ml-core-6.11.5-1.el9.elrepo.x86\_64 1/3

Установка : kernel-ml-modules-6.11.5-1.el9.elrepo.x86\_64 2/3

Запуск скриптлета: kernel-ml-modules-6.11.5-1.el9.elrepo.x86\_64 2/3

Установка : kernel-ml-6.11.5-1.el9.elrepo.x86\_64 3/3

Запуск скриптлета: kernel-ml-core-6.11.5-1.el9.elrepo.x86\_64 3/3

Запуск скриптлета: kernel-ml-6.11.5-1.el9.elrepo.x86\_64 3/3

Проверка : kernel-ml-6.11.5-1.el9.elrepo.x86\_64 1/3

Проверка : kernel-ml-core-6.11.5-1.el9.elrepo.x86\_64 2/3

Проверка : kernel-ml-modules-6.11.5-1.el9.elrepo.x86\_64 3/3

Установлен:

kernel-ml-6.11.5-1.el9.elrepo.x86\_64 kernel-ml-core-6.11.5-1.el9.elrepo.x86\_64

kernel-ml-modules-6.11.5-1.el9.elrepo.x86\_64

Выполнено!

Делаем новое ядро загружаемым по умолчанию

**grubby --default-kernel**

Получаем

/boot/vmlinuz-6.11.5-1.el9.elrepo.x86\_64

После перезагрузки системы проверяем версию ядра

**reboot**

**uname -sr**

Получаем

Linux 6.11.5-1.el9.elrepo.x86\_64

Новое ядро установлено.

Часть пятая. Установка KVM виртуальных машин.

Это виртуализация с помощью KVM (виртуальная машина на базе ядра) + QEMU.

Для этого требуется, чтобы центральный процессор вашего компьютера имел функцию Intel VT или AMD-V. Оригинал этой инструкции можно посмотреть здесь <https://www.server-world.info/en/note?os=AlmaLinux_9&p=kvm&f=1>

Интересную статья по KVM можно почитать здесь <https://habr.com/ru/articles/168791/>

Подключаемся к серверу по SSH, переходим на пользователя root

**su -**

Если не создали ранее, то создаем папку для дистрибутивов и обмены с клиентами Windows. Меняем группу доступа к этой папке на пользователя «user». Даем полные права для этой папки.

mkdir /home/distrib

chgrp -R user /home/distrib

chmod 777 /home/distrib

ls -l /home

Должно получиться для папки distrib

drwxrwxrwx. 2 root user 6 окт 24 12:33 distrib

Если ранее не сделали, то загружаем туда iso образ системы с сайта almalinux. Сделаем это командой «**curl**». Загрузка может занять 10 и более минут.

Ссылка может измениться, можно скопировать другую с сайта https://almalinux.org/get-almalinux/

**curl -o /home/distrib/AlmaLinux-9.4-x86\_64-dvd.iso https://repo.almalinux.org/almalinux/9.4/isos/x86\_64/AlmaLinux-9.4-x86\_64-dvd.iso**

Получаем

% Total % Received % Xferd Average Speed Time Time Time Current

Dload Upload Total Spent Left Speed

100 10.0G 100 10.0G 0 0 15.5M 0 0:11:02 0:11:02 --:--:-- 19.3M

Проверяем наличие скаченного файла

**ls -lh /home/distrib**

Должно получиться примерно так

-rwxr--r--. 1 qemu qemu 11G ноя 11 10:08 AlmaLinux-9.4-x86\_64-dvd.iso

Устанавливаем необходимые пакеты, если они не были установлены при первичной установке системы

**dnf install -y qemu-kvm libvirt virt-install bridge-utils libvirt-daemon-kvm**

Включаем автозапуск сервиса и запускаем его. Проверяем работу сервиса

**systemctl enable --now libvirtd**

**systemctl status libvirtd**

Проверяем поддержку виртуализации наши сервером

[**lsmod**](https://www.server-world.info/en/command/html/lsmod.html)**| [grep](https://www.server-world.info/en/command/html/grep.html) kvm**

Получаем примерно так

kvm\_intel 348160 0

kvm 1056768 1 kvm\_intel

irqbypass 16384 1 kvm

Переходим к настройке сетевого бриджа для ВМ. Я буду настраивать режим бриджа при котором наши ВМ будут получать IP адрес в этой же локальной сети, что и основной сервер. Поэтому одна сетевая плата будет занята этим бриджем и не может использоваться основным сервером. При таком способе подключения на сервере нужно иметь не менее двух сетевых плат.

Если у вас на сервере нет двух плат, то добавьте еще одну или используйте другой способ настройки сети для ВМ.

Задаем имя серверу (вместо smsrv.h68 надо вставить ваше имя сервера)

**hostnamectl set-hostname smsrv.h68**

Перезагружаем сервис systemd-hostnamed для применения изменений

**systemctl restart systemd-hostnamed**

Проверяем имя сервера

**hostname -f**

Должно получиться **smsrv.h68**

Выводим информацию о сетевых подключениях

**nmcli device status**

Получаем

DEVICE TYPE STATE CONNECTION

eno1 ethernet подключено eno1

eno2 ethernet подключено eno2

enp0s20f0u5u2c2 ethernet подключено enp0s20f0u5u2c2

lo loopback подключено (внешнее) lo

Настраивать будем первый интерфейс eno1, поэтому, если вы подключены удаленно,

то подключитесь к машине через второй интерфейс eno2 или поменяйте имя интерфейса в настройках ниже на имя второго интерфейса.

ВНИМАНИЕ! В следующих командах замените интерфейс с именем eno1 на имя вашего интерфейса.

Добавляем бридж [br0]

**nmcli connection add type bridge autoconnect yes con-name br0 ifname br0**

Получаем

Подключение «br0» (fc3188c8-cb86-47e6-8c0c-20e20af7ba1a) успешно добавлено.

Задаем IP address для [br0]

**nmcli connection modify br0 ipv4.addresses 10.0.0.30/24 ipv4.method manual**

Задаем шлюз для [br0]

**nmcli connection modify br0 ipv4.gateway 10.0.0.1**

Задаем DNS для [br0]

**nmcli connection modify br0 ipv4.dns 10.0.0.10**

Задаем DNS search base для [br0]. Здесь вместо **smsrv.h68** надо вставить имя вашего сервера (посмотреть можно командой «hostname -f»)

**nmcli connection modify br0 ipv4.dns-search smsrv.h68**

Привязываем сеть ВМ к интерфейсу eno1.

Удаляем текущий интерфейс (замените eno1 на имя вашего интерфейса)

**nmcli connection del eno1**

Добавляем удаленный интерфейс в качестве элемента [br0]

**nmcli connection add type bridge-slave autoconnect yes con-name eno1 ifname eno1 master br0**

Выводим информацию о сетевых подключениях

**nmcli device status**

Получаем

DEVICE TYPE STATE CONNECTION

eno2 ethernet подключено eno2

br0 bridge подключено br0

enp0s20f0u5u2c2 ethernet подключено enp0s20f0u5u2c2

eno1 ethernet подключено eno1

lo loopback подключено (внешнее) lo

И другой командой

**ip addr**

Получаем

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid\_lft forever preferred\_lft forever

inet6 ::1/128 scope host

valid\_lft forever preferred\_lft forever

2: eno1: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc mq master br0 state UP group default qlen 1000

link/ether 7c:c2:55:83:19:08 brd ff:ff:ff:ff:ff:ff

altname enp62s0f0

3: eno2: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc mq state UP group default qlen 1000

link/ether 7c:c2:55:83:19:09 brd ff:ff:ff:ff:ff:ff

altname enp62s0f1

inet 192.168.9.147/24 brd 192.168.9.255 scope global dynamic noprefixroute eno2

valid\_lft 80704sec preferred\_lft 80704sec

inet6 fe80::7ec2:55ff:fe83:1909/64 scope link noprefixroute

valid\_lft forever preferred\_lft forever

4: enp0s20f0u5u2c2: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc fq\_codel state UNKNOWN group default qlen 1000

link/ether be:3a:f2:b6:05:9f brd ff:ff:ff:ff:ff:ff

inet 169.254.3.1/24 brd 169.254.3.255 scope link dynamic noprefixroute enp0s20f0u5u2c2

valid\_lft 858299sec preferred\_lft 858299sec

inet6 fe80::bc3a:f2ff:feb6:59f/64 scope link noprefixroute

valid\_lft forever preferred\_lft forever

7: br0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000

link/ether 7c:c2:55:83:19:08 brd ff:ff:ff:ff:ff:ff

inet 192.168.9.146/24 brd 192.168.9.255 scope global dynamic noprefixroute br0

valid\_lft 85015sec preferred\_lft 85015sec

inet6 fe80::92ea:7123:a3eb:9489/64 scope link noprefixroute

valid\_lft forever preferred\_lft forever

8: vnet0: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc noqueue master br0 state UNKNOWN group default qlen 1000

link/ether fe:54:00:2d:33:ae brd ff:ff:ff:ff:ff:ff

inet6 fe80::fc54:ff:fe2d:33ae/64 scope link

valid\_lft forever preferred\_lft forever

Перегружаем сервер

**reboot**

Далее необходимо подключиться к серверу напрямую или удаленно по RDP. Это необходимо, потому что установку ВМ будем производить в графическом режиме.



Так как на форуме не подсказали по необходимым параметрам ВМ, то будем создавать ВМ с памятью 8192 ГБ, процессором 4 ядра, размером диска 30 ГБ (память и количество ядер можно будет впоследствии изменить). Как настроить ВМ без командной строки при помощи «Менеджера виртуальных машин» можно прочитать здесь <https://www.server-world.info/en/note?os=AlmaLinux_9&p=kvm&f=3>

В меню внизу экрана запускаем терминал, переключаемся на пользователя «root».

И выполняем следующую команду:

virt-install \

--name=vm \

--vcpus=4 \

--memory=8192 \

--disk size=30,format=qcow2 \

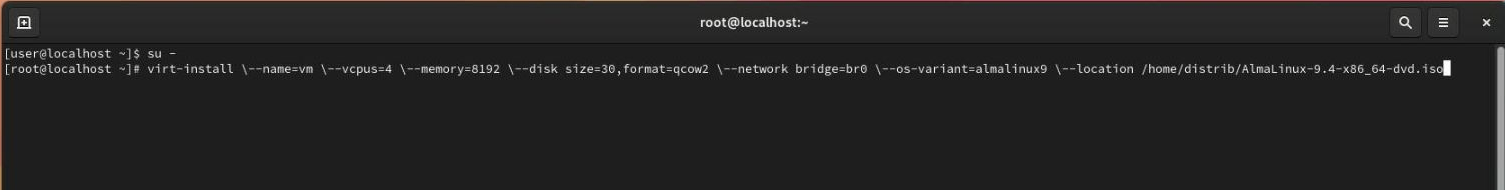
--network bridge=br0 \

--os-variant=almalinux9 \

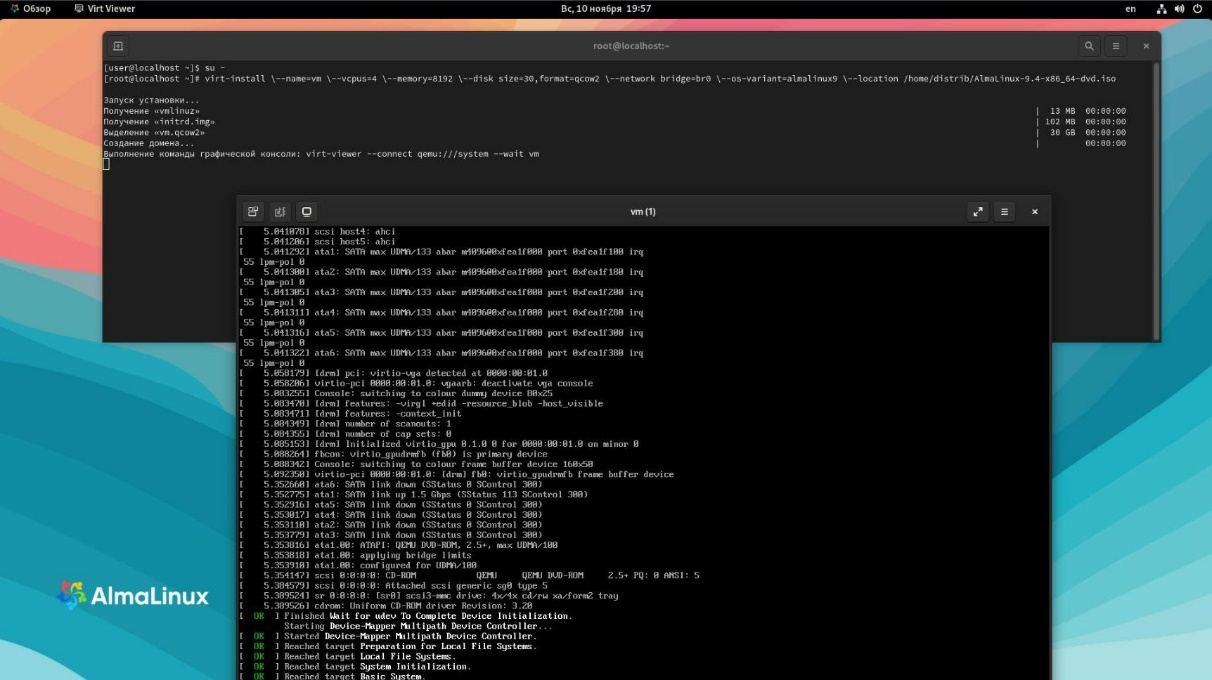
--location /home/distrib/AlmaLinux-9.4-x86\_64-dvd.iso

Проще эту команду ввести одной строкой:

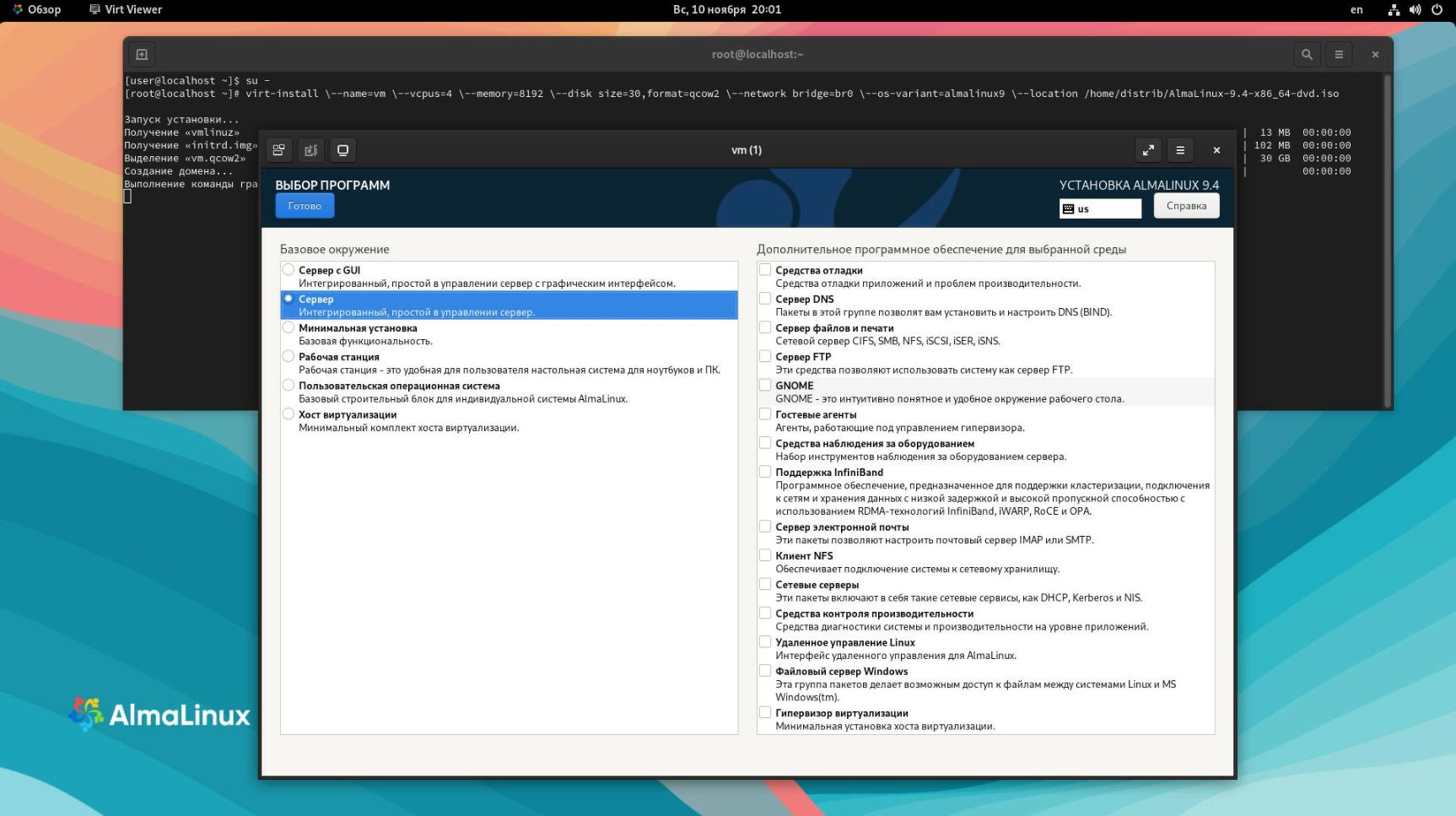
**virt-install \--name=vm \--vcpus=4 \--memory=8192 \--disk size=30,format=qcow2 \--network bridge=br0 \--os-variant=almalinux9 \--location /home/distrib/AlmaLinux-9.4-x86\_64-dvd.iso**

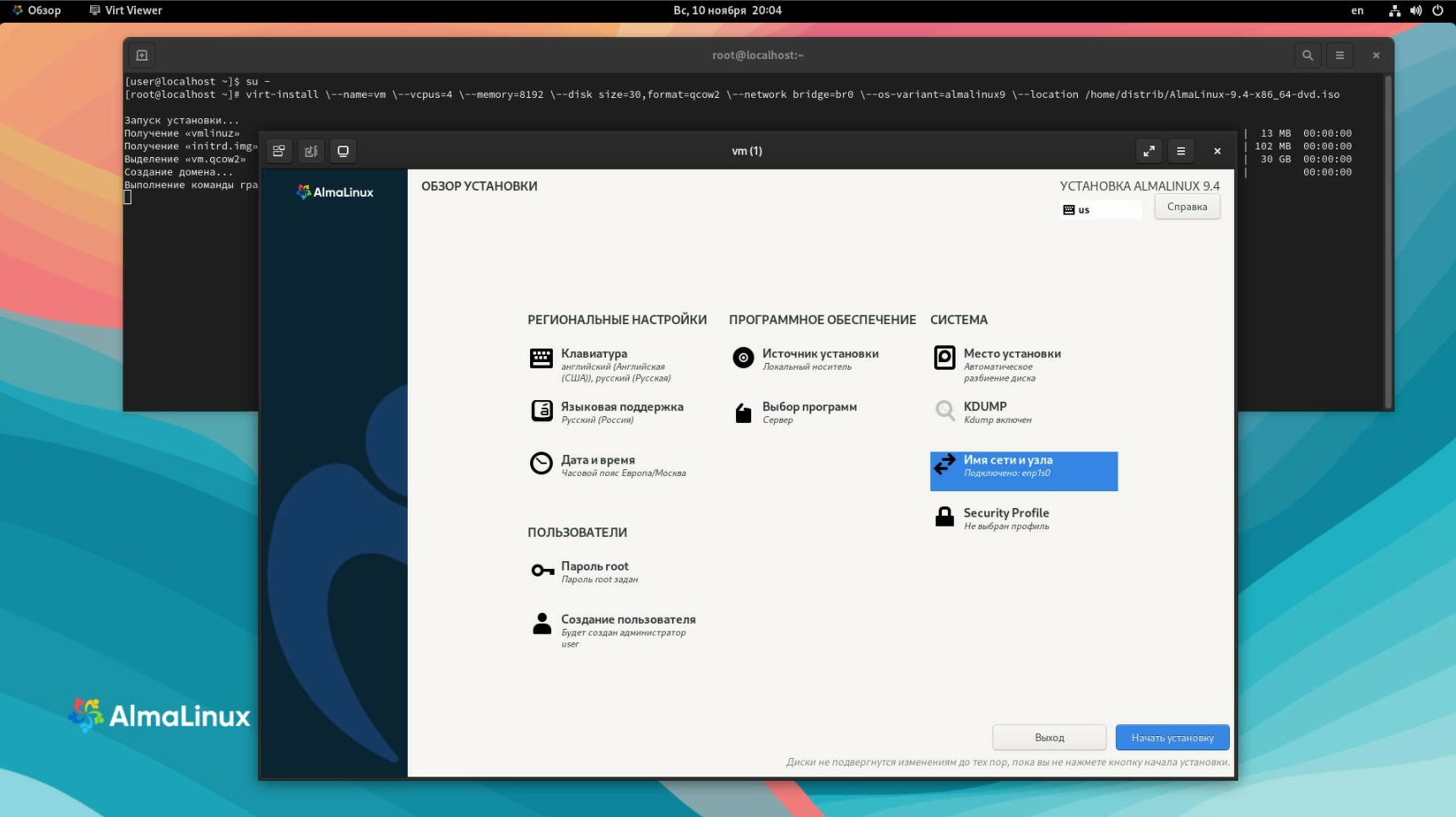


Если сделали все правильно, то начнется процесс создания ВМ

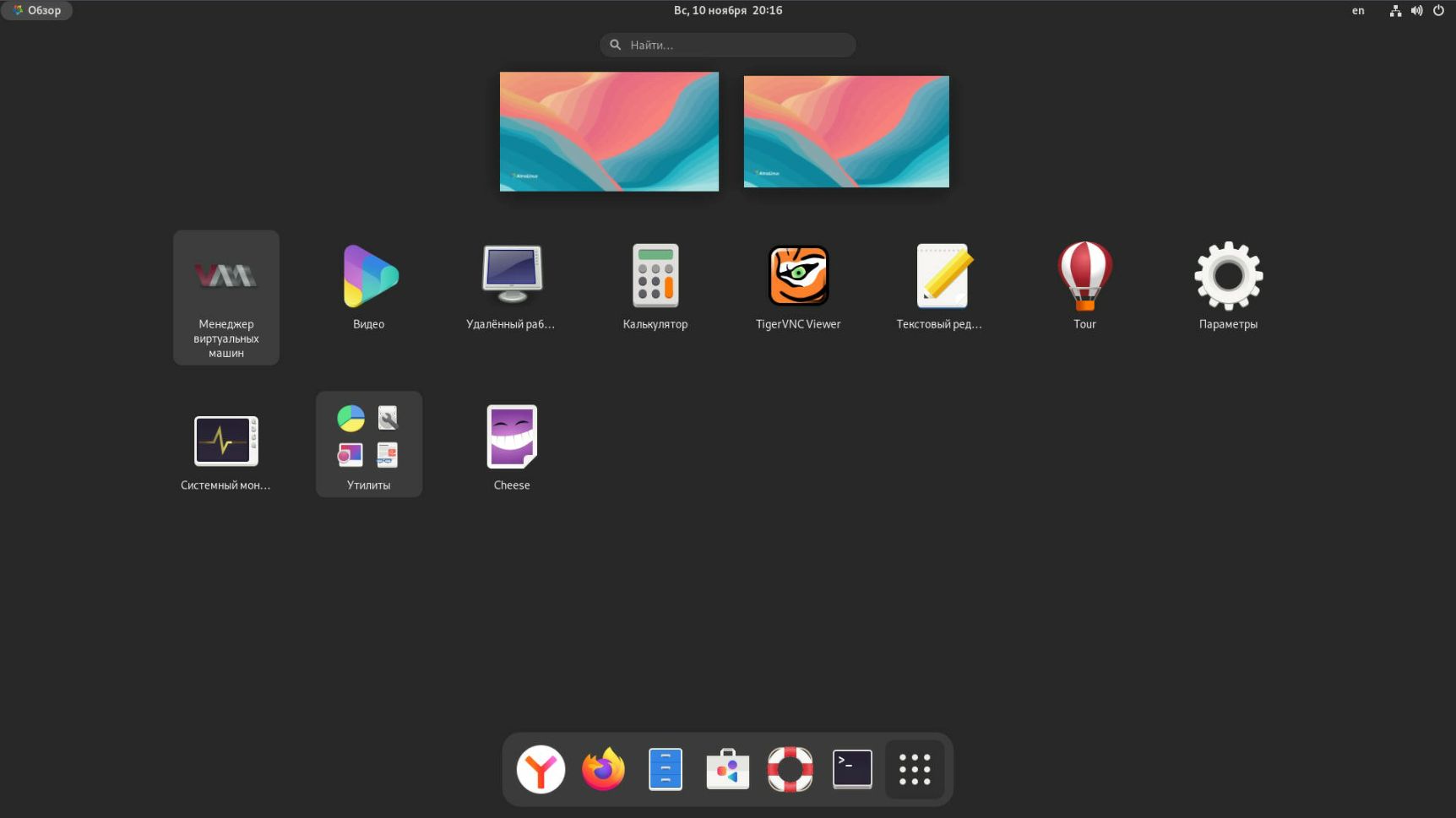


Далее откроется графическое меню настройки нашей ВМ. Процесс настройки можно посмотреть в первой части этой инструкции. Только сейчас мы выбираем установку сервера без GUI (без графической оболочки) и в дополнительных программах ничего не выбираем. Если понадобится, то это можно будет сделать позже, при настройке конкретной ВМ.

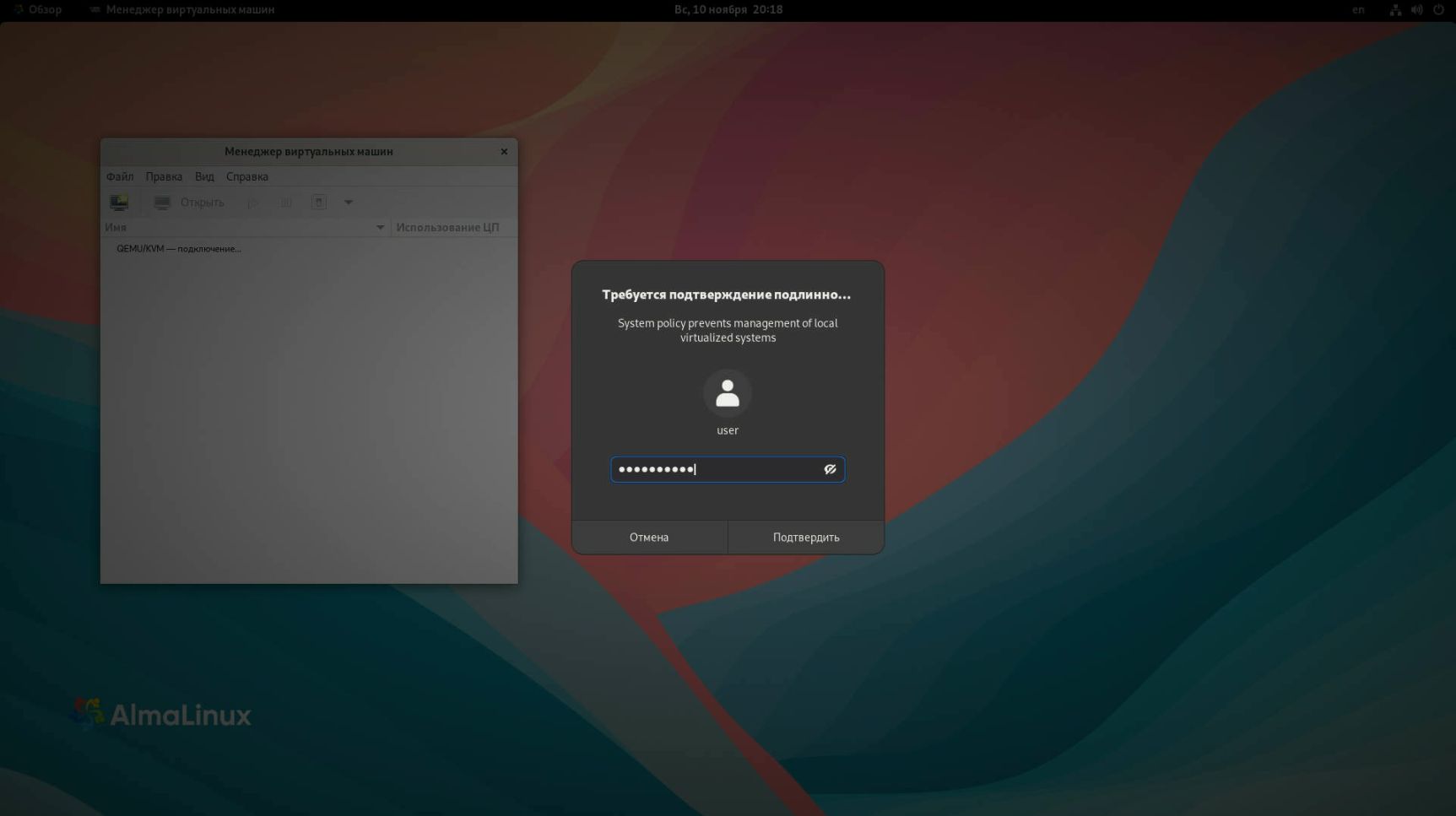




После установки перегружаем созданную ВМ. Активируем «Cockpit». Далее в меню нажимаем правую иконку и запускаем «Менеджер виртуальных машин». Если меню не отображается на экране, то нажимаем в левом верхнем углу кнопку «Обзор».

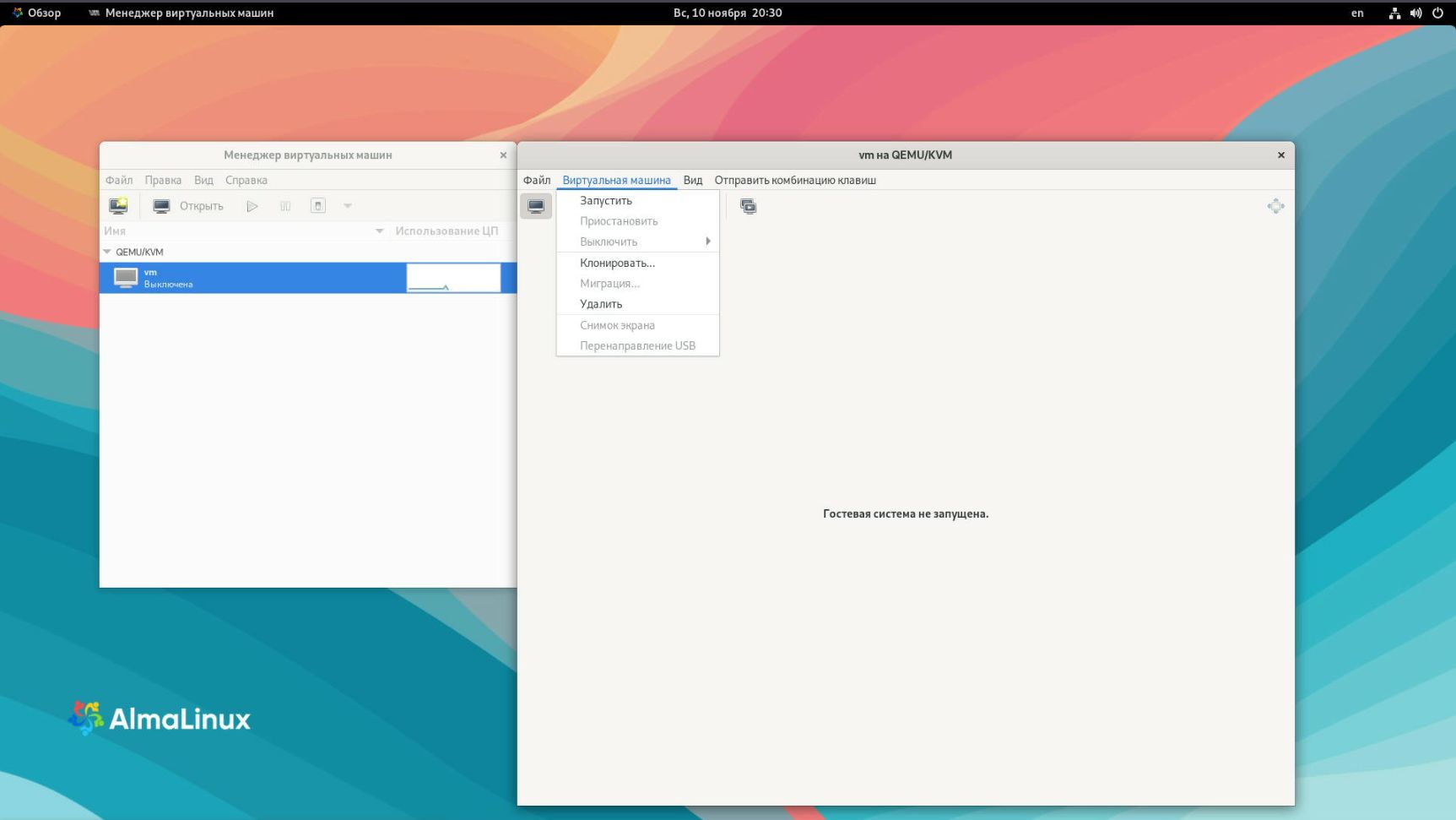


Открывается окно с запросом пароля, вводим свой пароль пользователя «user»

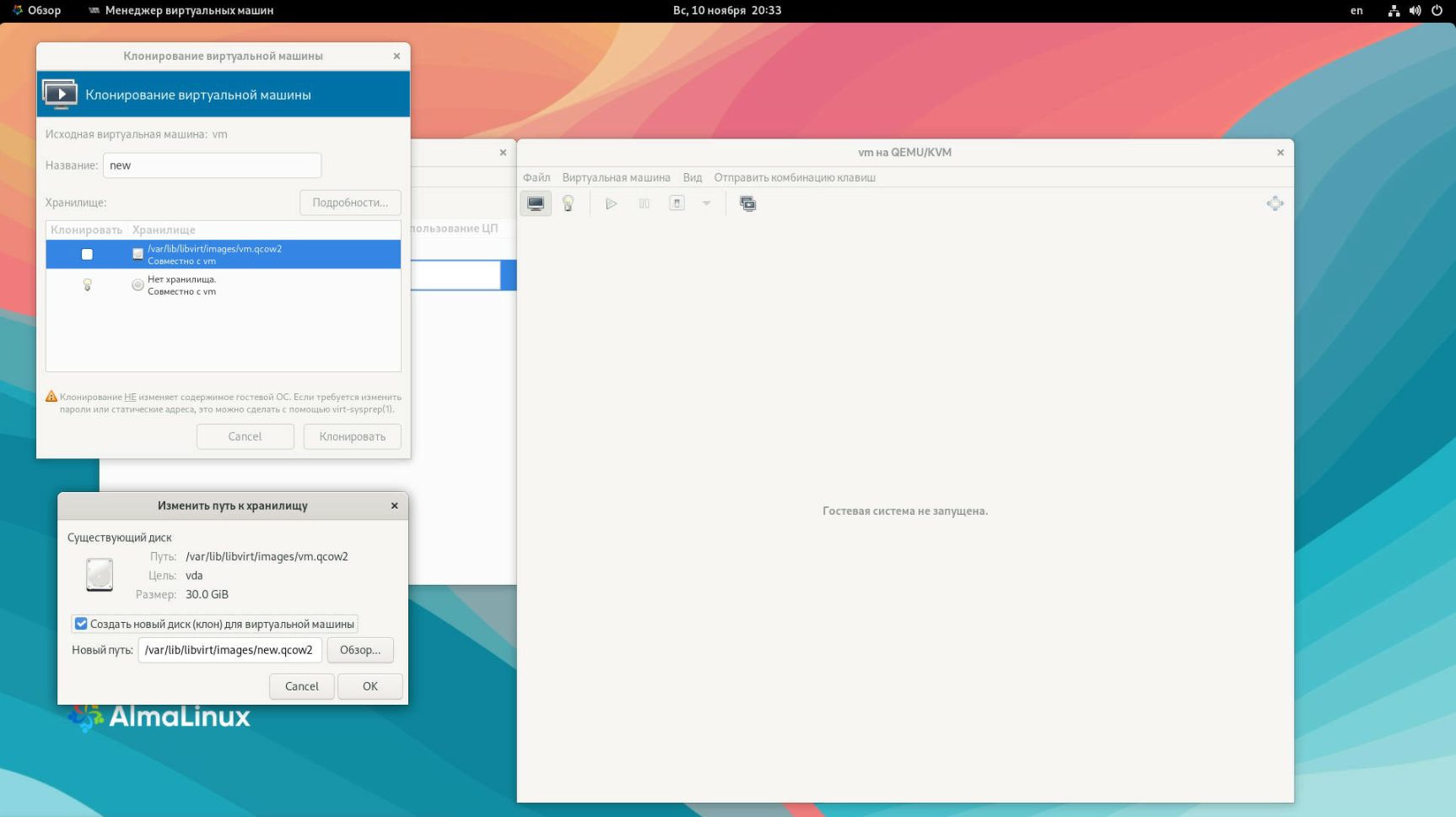


Выключаем нашу ВМ и нажимаем «Клонировать». Делать будем два клона. Один под названием «new», он будет шаблоном который мы не будем изменять и второй «test-vm». Из этого клона, после некоторой подготовки, будем создавать наши рабочие ВМ. Ниже я объясню почему мы так делаем.

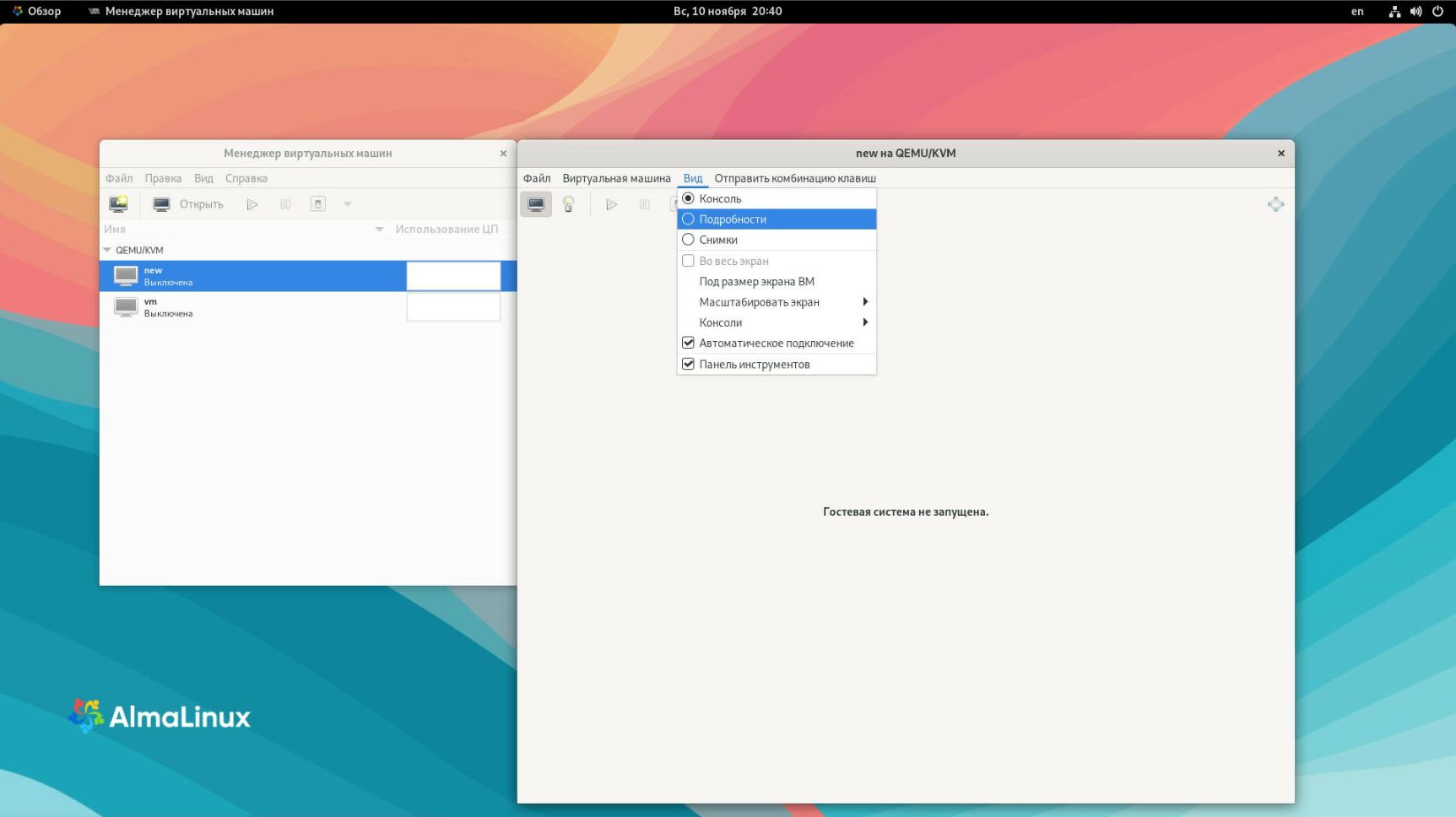
Также операцию клонирования ВМ можно сделать через интерфейс «Cockpit». Попробовать можете самостоятельно. Этот вариант, как мне кажется, будет более простым.

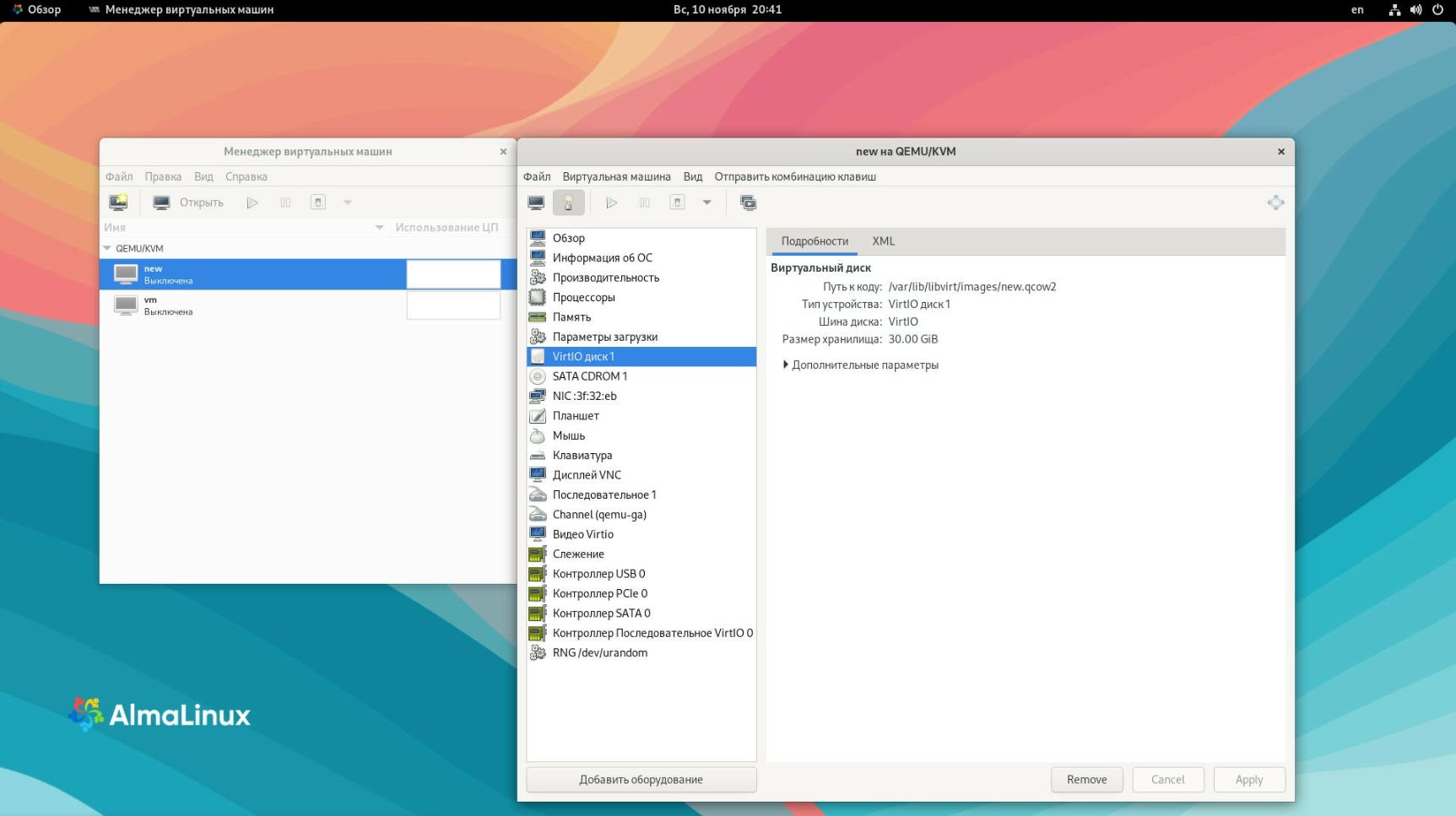


Клонируем первую, а затем точно так же вторую



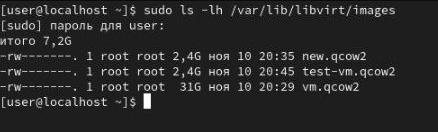
Проверяем параметры нашей новой ВМ. Выбираем ВМ, нажимаем «Открыть» и на закладке «Вид» выбираем «Подробности»



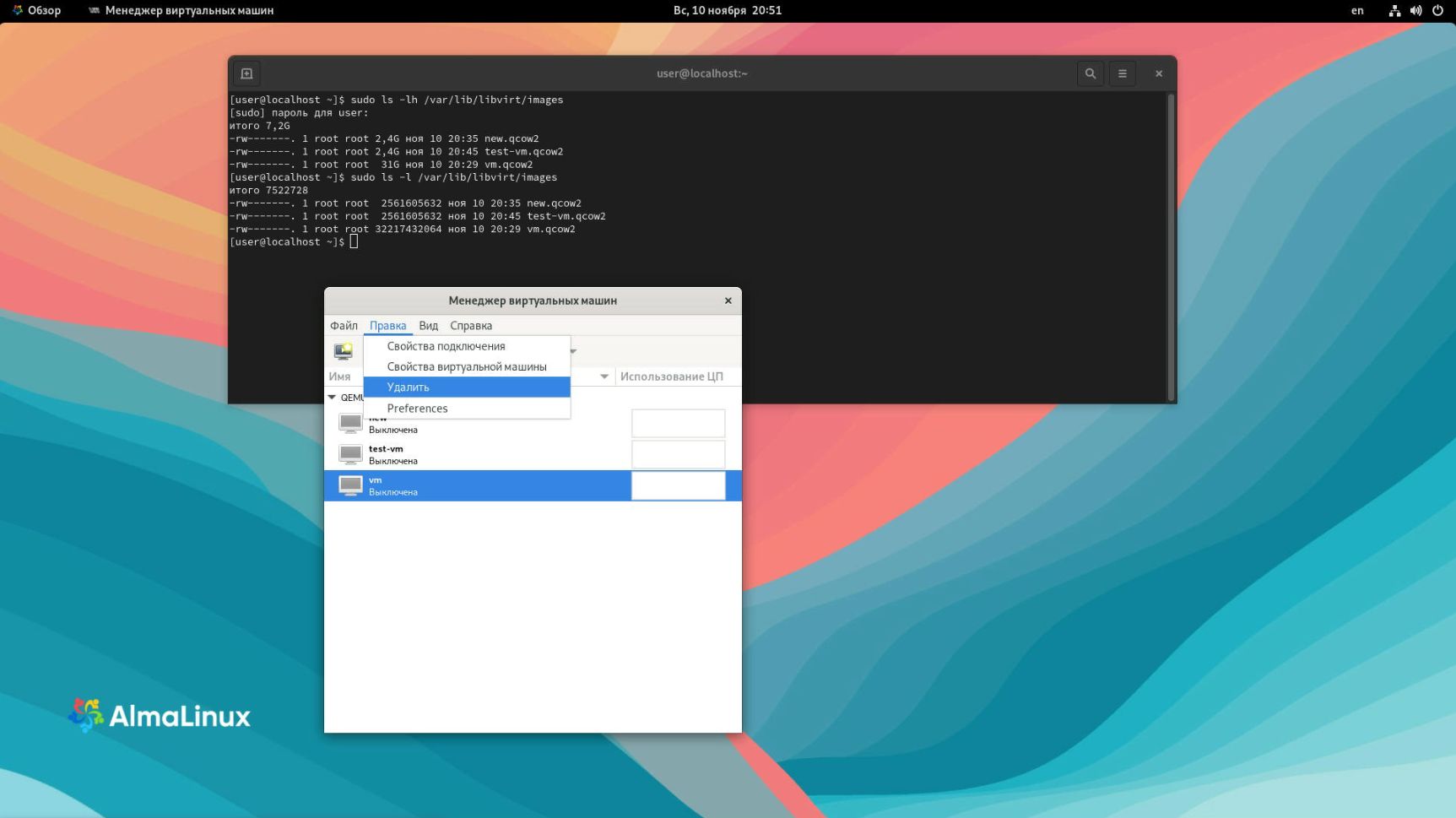


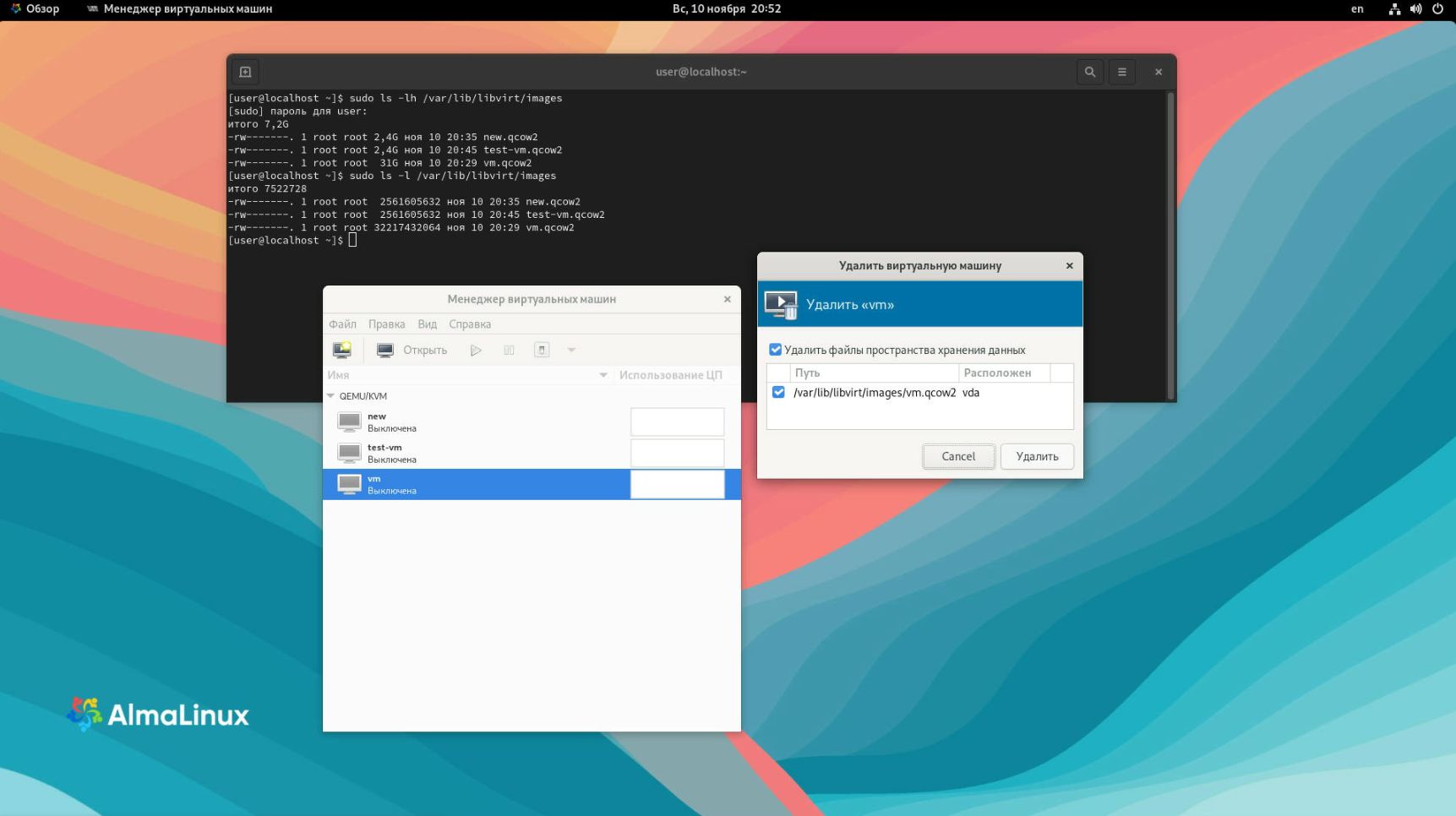
А теперь я объясню зачем мы делали два клона. Посмотрим размеры наших ВМ. По умолчанию они находятся в /var/lib/libvirt/images. В терминале выполняем команду (аналог dir для Windows):

**sudo ls -lh /var/lib/libvirt/images**

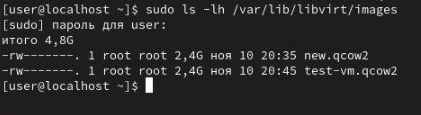


Мы видим, что размеры наших клонов по 2,4G, а созданная нами 31G. При создании новой ВМ под нее сразу резервируется все место согласно ее размеру, а при клонировании занимается только место согласно ее текущему размеру. По мере установки компонентов в эти ВМ этот размер будет увеличиваться. Теперь мы удаляем нашу первую ВМ и освобождаем место на диске.



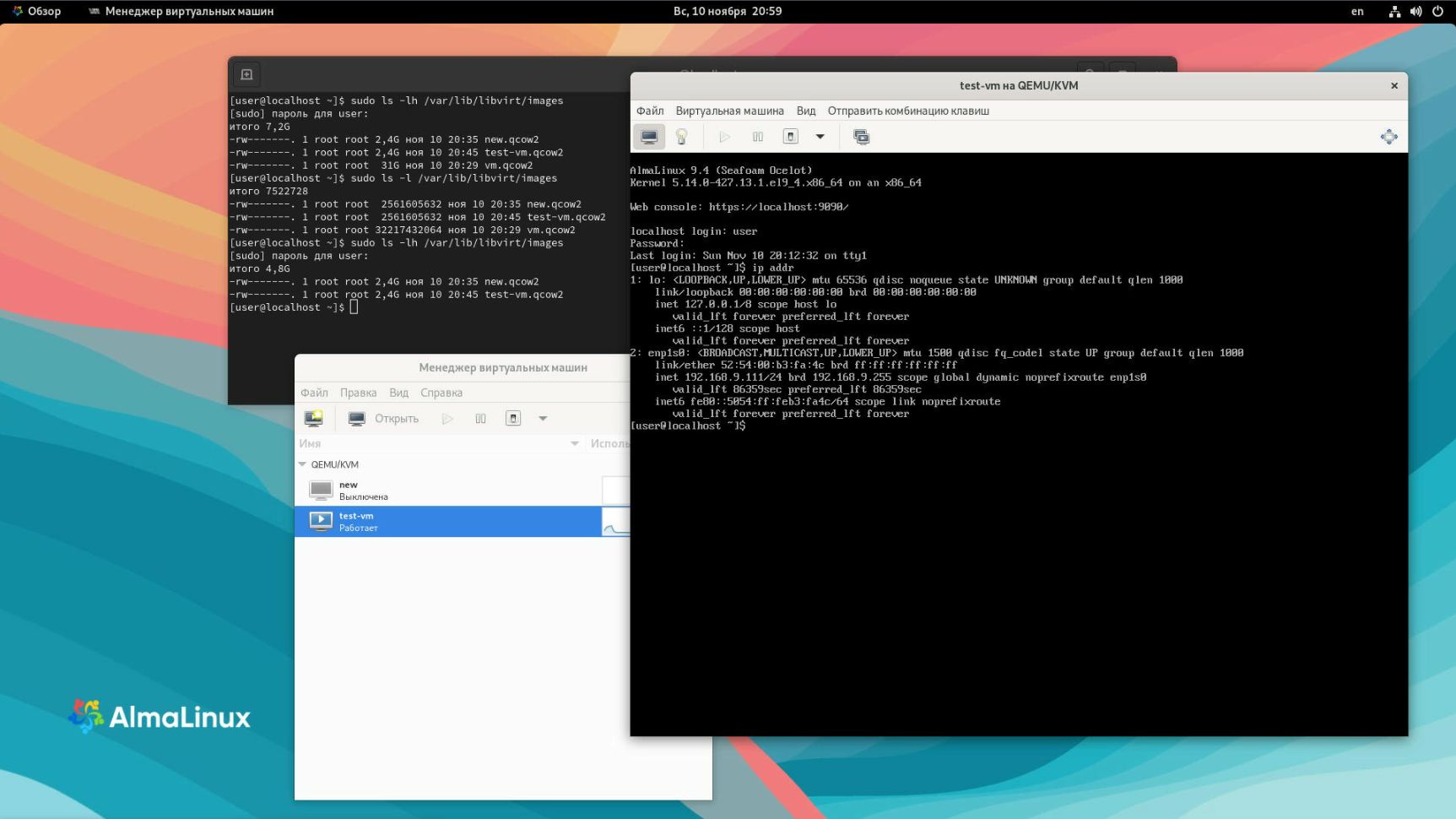


Проверяем снова расположение наших ВМ

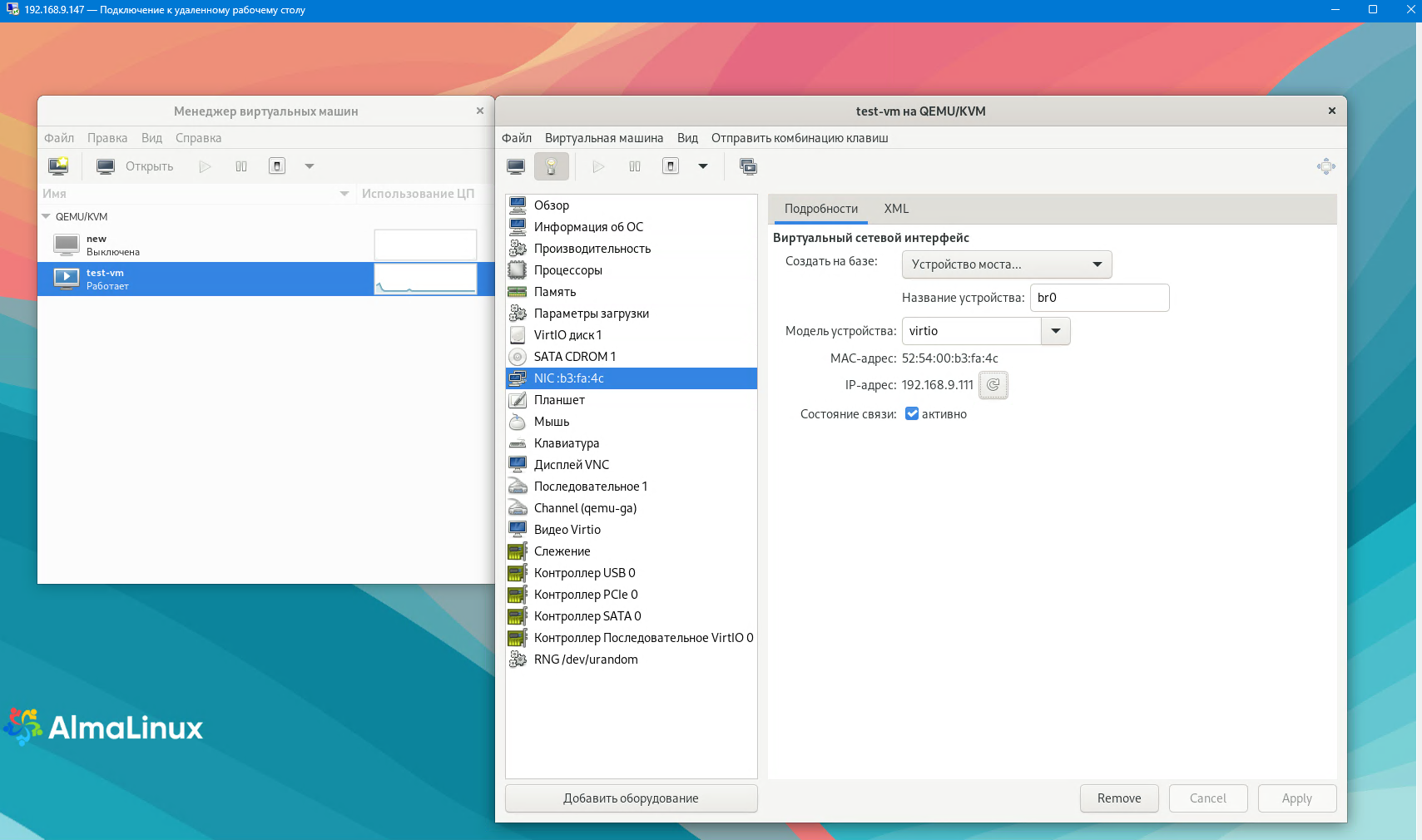


Теперь переходим к настройке наше ВМ «test-vm». В менеджере ВМ открываем ее и запускаем. Выполняем команду и смотрим какой у нее ip адрес.

**ip addr**

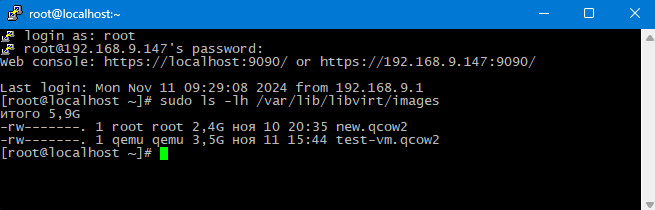
****

Или в настройках через «Менеджер виртуальных машин»

****

У меня адрес получился 192.168.9.111. Далее нужно провести первоначальную настройку ВМ, чтобы потом не повторять эти операции для наших рабочих ВМ. Будем клонировать наши рабочие машины с этой ВМ.

Выполняем команду «**exit»**, затем закрываем «Менеджер виртуальных машин». ВМ остается запущенной. Настраивать ВМ будем удаленно, подключаемся к ВМ через Putty и повторяем наш процесс настройки согласно частям 2 и 4 этой инструкции. **Подключение по Xrdp и графический интерфейс настраивать не будем.** Проверяем размер нашей ВМ после этих манипуляций. Видим, что размер увеличился (проверять нужно на основном сервере у меня он имеет ip 192.168.9.147).



После этих настроек путем клонирования с этой ВМ делаем три ВМ например «tegu», «postgre» и «web». В настройках через «Менеджер виртуальных машин» или «Cockpit» включаем их в автозагрузку, проверяем выделенные им ip адреса и, если нужно, изменяем их на свои. Изменить ip адреса можно просто подключившись к этим машинам через «Cockpit» [**https://ip\_vm:9090**](https://ip_vm:9090)

Или через командную строку (у себя настроил так для одной из ВМ), имя интерфейса enp1s0 надо заменить на свое.

Задаем ip адрес для [enp1s0]

**nmcli connection modify enp1s0 ipv4.addresses 192.168.9.151/24 ipv4.method manual**

Задаем шлюз для [enp1s0]

**nmcli connection modify enp1s0 ipv4.gateway 192.168.9.1**

Задаем DNS для [enp1s0]

**nmcli connection modify enp1s0 ipv4.dns 192.168.9.1**

Это еще можно сделать через редактирование файла конфигурации (сам не проверял)

**nano /etc/NetworkManager/system-connections/enp1s0.nmconnection**

В секции [ipv4] прописать следующее

[ipv4]

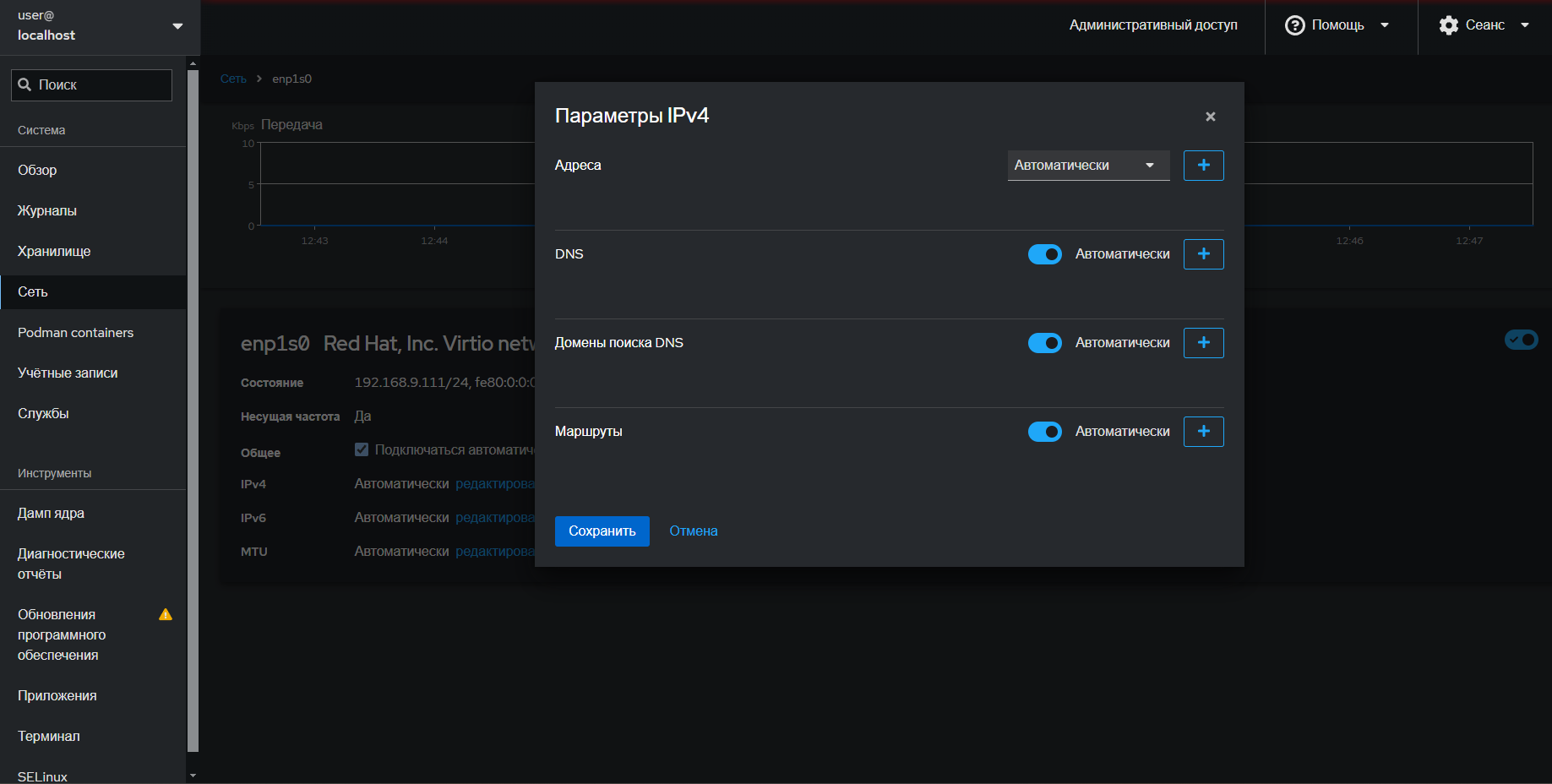
address1=192.168.9.151/24,192.168.9.1

dns=192.168.9.1;

method=manual

Настройки применятся после перезагрузки ВМ или после перезапуска службы «NetworkManager» (**вы потеряете связь с ВМ и нужно будет подключиться по новому ip**)

**systemctl restart NetworkManager**



На этом первоначальная подготовка наших ВМ для почтового сервера завершена. Индивидуальные настройки каждой ВМ рассмотрим в следующих частях инструкции.